AD-A048 024 WEAPONS RESEARCH ESTABLISHMENT SALISBURY (AUSTRALIA) F/G 9/2
AN IBM 370/168 SOFTWARE PACKAGE FOR PREDICTING THE STABILITY OF--ETC(U)
SEP 77 H M JOHNSON

UNCLASSIFIED WRE-TR-1882(A) NL

(14) WRE-TR-1882(A)

(12)

AR-000-945

AD A048024

AUSTRALIA

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

WEAPONS RESEARCH ESTABLISHMENT

SALISBURY, SOUTH AUSTRALIA

(9) TECHNICAL REPORT, 1882 (A)

(6) AN IBM 370/168 SOFTWARE PACKAGE FOR PREDICTING THE STABILITY OF A PROPOSED HYBRID SIMULATION,

(10) H.M. JOHNSON

(11) Sep 77

(12) 98p.

AD No. DDC FILE COPY

Approved for Public Release.

SEPTEMBER 1977

COPY No. 20

371700

AR-000-945

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

WEAPONS RESEARCH ESTABLISHMENT

TECHNICAL REPORT 1882 (A)

AN IBM 370/168 SOFTWARE PACKAGE FOR PREDICTING THE STABILITY OF A PROPOSED HYBRID SIMULATION

H.M. Johnson

SUMMARY

Time delays and sample-and-hold operations in hybrid computation can cause serious stability problems, with the difficulty that considerable time and effort is wasted if this instability is not known in advance of programming the hybrid computer. A method of analysis exists which can be used to predict the stability of a given computing scheme. Although many problems can be solved by direct analysis, large-sized problems present some difficulty.

This report describes a software package which implements the analysis procedure. The software package provides a quick and flexible means of determining in advance of programming the stability of proposed partitioning schemes for large or small-sized problems. The method is applicable for certain classes of systems, viz those which can be linearized about a current state of motion. This can, however, be realized for a wide variety of systems, since it is not required to obtain an accurate simulation of system behaviour, only to approximate the speed of response sufficiently well for the purposes of stability analysis.

POSTAL ADDRESS: The Director, Weapons Research Establishment, Box 2151, G.P.O., Adelaide, South Australia, 5001.

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS	2 SECURITY CLASSIFICATION
AR Number: AR-000-945	a. Complete Document: Unclassified
Report Number: WRE-TR-1882(A)	b. Title in Isolation: Unclassified
Other Numbers:	c. Summary in Isolation: Unclassified

3 TITLE: AN IBM 370/168 SOFTWARE PACKAGE FOR PREDICTING THE STABILITY OF A PROPOSED HYBRID SIMULATION

4 PERSONAL AUTHOR(S):

H.M. Johnson

5 DOCUMENT DATE: September 1977

6 6.1 TOTAL NUMBER OF PAGES: 95

6.2 NUMBER OF REFERENCES: 4

7 7.1 CORPORATE AUTHOR(S):

Weapons Research Establishment

7.2 DOCUMENT (WING) SERIES AND NUMBER

Applied Physics Wing TR-1882

8 REFERENCE NUMBERS

a. Task: DST 76/04

b. Sponsoring Agency: DST

9 COST CODE: 341856

10 IMPRINT (Publishing establishment):

Weapons Research Establishment

11 COMPUTER PROGRAM(S) (Title(s) and language(s))

12 RELEASE LIMITATIONS (of the document):

Approved for Public Release

12.0	OVERSEAS	NO		P.R.	1	A		B		C		D		E	

Security classification of this page: UNCLASSIFIED

Security classification of this page: UNCLASSIFIED

13 ANNOUNCEMENT LIMITATIONS (of the information on these pages):

No limitation

14 DESCRIPTORS:

a. EJC Thesaurus Terms	Computer systems programs Computation Predictions Stability Simulation Computerized simulation
b. Non-Thesaurus Terms	Hybrid computation

15 COSATI CODES:

0902

16 LIBRARY LOCATION CODES (for libraries listed in the distribution):

SW SR SD AACA NL MG

17 SUMMARY OR ABSTRACT:
(if this is security classified, the announcement of this report will be similarly classified)

Time delays and sample-and-hold operations in hybrid computation can cause serious stability problems, with the difficulty that considerable time and effort is wasted if this instability is not known in advance of programming the hybrid computer. A method of analysis exists which can be used to predict the stability of a given computing scheme. Although many problems can be solved by direct analysis, large-sized problems present some difficulty.

This report describes a software package which implements the analysis procedure. The software package provides a quick and flexible means of determining in advance of programming the stability of proposed partitioning schemes for large or small-sized problems. The method is applicable for certain classes of systems, viz those which can be linearized about a current state of motion. This can, however, be realized for a wide variety of systems, since it is not required to obtain an accurate simulation of system behaviour, only to approximate the speed of response sufficiently well for the purposes of stability analysis.

Security classification of this page: UNCLASSIFIED

TABLE OF CONTENTS

Page No.

LIST OF APPENDICES

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Whenever a simulation problem is to be programmed for hybrid computation, the first question to be answered is how should the problem be partitioned between the analogue and digital portions of the hybrid computer? One naturally leans towards a scheme which is convenient for programming, and which allows easy change of system parameters and equations during a series of runs. The complicating factor is that due to the effects of time delays caused by calculations on the digital computer, and the sample-and-hold operations of the interfaces between the two computers, the chosen configuration may not be mathematically stable. Such time delays are always present in a true hybrid simulation, and it would be useful to know, in advance of actually programming a configuration, what time delays (i.e. digital computer step-length) can be tolerated within the proposed partitioning scheme, if the simulation is to run in real time. A considerable amount of time and effort may be wasted in partitioning and programming a problem on a trial and error basis.

Reference 1 provides a method of analysis which will allow the user to determine, in advance of programming, the stability or otherwise of a given hybrid simulation for various step-lengths, provided the user can find a linear approximation to the response of the system under consideration. One of the recommendations of reference 1, viz, that the stability analysis procedure addressed in the report be implemented as a software package, has now been completed.

The material in this report is a description of the software package, including information required by a potential user. Section 5 deals with two simulation problems, and shows the method by which the linear representations of the systems are defined in a form suitable for input to the software package. One of these examples is further considered with a much increased speed of response. This subsequently allows the question of stability of the modifed system to be settled by theoretical analysis. This example provides a useful test case for the software package. In addition, one of the applications of Section 5 has been checked by hybrid simulation giving good agreement.

2. BRIEF OVERVIEW OF THE HYBRID PROBLEM

Once a user of the hybrid computer has decided upon a partitioning scheme, he needs to answer two further questions. The first is how long will the digital portion of the hybrid computer take to compute those parts of the simulation problem assigned to it? The time for computation, h, provides a minimum value for the simulation step-length. The scheme must be stable for at least this step-length, and since there are other delays, such as sample-and-hold delays associated with hybrid computation(ref.2), it is clear that the step-length will need to be at least

$$h_s = h + \delta h\, s$$

where δh represents the accumulation, per step, of these additional delays.

The software package provides a quick and flexible means of determining the answer to the second problem, viz, will the proposed partitioning scheme be stable for the minimum allowable step-length h_s? In practice, the second question is the one to be addressed first.

Once a system has been linearized it is possible, for little further effort, to examine a number of different partitioning schemes with repeated use of the software package. Only after careful selection of a scheme from the possible contenders is it time to address the first question posed above, and then usually only by specialists familiar with the particular hybrid installation. It may turn out that the only way to achieve an accurate enough computation time is to program the equations assigned to the digital computer and run them on the digital portion of the hybrid installation. Sample-and-hold delays must be added to the

computation time, before the user is in a position to establish whether, h_s, the minimum step-length for the hybrid simulation, is within the step-length for mathematical stability.

The method of determining the stability of a system reduces to determining the stability, or otherwise, of integration processes. As such, the method is applicable only for certain classes of systems, namely those which can be linearized about a current state of motion. However, this can be realized for a great many systems of interest, since it is not required to provide an accurate simulation of system behaviour, only to approximate the speed of response sufficiently well for the purposes of stability analysis.

Appendix I, and further analysis given in reference 1, show that the stability requirements for a given scheme of partitioning can be derived in a straightforward manner. The ease with which this can be done, however, belies the care needed in the selection of an acceptable scheme. For although the stability analysis procedure described in Appendix I may show that a particular configuration is stable, the stability may be only marginal. Since we are employing a linear approximation to the real simulation model, the solution for the latter may therefore turn out to be unstable. In particular, it should be recognized that the partitioning schemes under consideration allow in principle for the part of the simulation on the analogue computer to be unstable, even though the overall simulation is not.

It was shown in reference 1 that relatively poor stability results whenever integrations are carried out digitally and derivatives are evaluated on the analogue computer, or derivatives are evaluated digitally and the remainder of the simulation is carried out on the analogue computer. The strong inference is that because of the relatively poor stability of either division, one should, wherever possible, integrate derivatives on the same portion of the hybrid computer as they are calculated. In particular, "fast" loops should be simulated on the analogue portion or if this is not possible, entirely on the digital computer.

This does not mean that one should rule out alternative schemes. In fact, the recommendation is that the stability analysis procedure described in Appendix I, and implemented in the software package, be used to determine the stability of partitioning schemes devised on the basis of convenience to the user. However, if there are no such requirements, the principles described in the last paragraph should be followed.

3. REQUIREMENTS OF THE SOFTWARE PACKAGE

This Section defines the partitioning problem mathematically, and lists the procedures required of a software solution for analysing stability of a system.

3.1 Problem definition

Following reference 1, we consider the problem of solving on the hybrid computer the set of differential equations

$$\dot{\underline{x}} = A\underline{x} \qquad (1)$$

which are obtained by linearization of the defining equations of the system to be simulated.

Figure 1 shows the general partitioning scheme for equation (1), in which integrations and evaluation of derivatives can be performed digitally or on the analogue without restriction. The analysis of the scheme is provided in Appendix I, with further explanation in reference 1.

The equations can be rewritten as

$$\dot{\chi} = A_1\chi + A_2\zeta + D_1\chi + D_2\zeta$$

$$\dot{\zeta} = A_3\chi + A_4\zeta + D_3\chi + D_4\zeta$$

where

$$\chi = \begin{pmatrix} \chi \\ \zeta \end{pmatrix}$$

Further, $\dot{\chi}$ is evaluated on the analogue portion and $\dot{\zeta}$ is evaluated digitally, and the vectors $A_1\chi$, $A_2\zeta$, $A_3\chi$, $A_4\zeta$ are computed on the analogue portion, while $D_1\chi$, $D_2\zeta$, $D_3\chi$, and $D_4\zeta$ are determined digitally.

In practice, for all but the most complex partitioning schemes, some of the matrices A_2, A_3, A_4, D_2, D_3 and D_4 will be null, and the others, including A_1 and D_1 will be sparse.

3.2 Four steps required of the software package

The main steps in the procedure for analysing the stability of the general partitioning problem are(see ref.1):-

(1) Specify the matrices A_1, ... A_4, D_1, ... D_4.

(2) Determine a fundamental matrix Φ for A_1 (following procedures described in Appendix I of reference 1) by two methods, providing a user option for either, namely

(i) the direct approach using eigenvalue theory, or

(ii) numerical integration over (o,h) for a given h.

(3) For a given value, or values of h, determing G(h) and ensure that G(h) corresponds to a stable analogue simulation, and Q(h), the latter by numerical integration over (o,h); G(h), Q(h) are described in Appendix I.

(4) Construct D and determine its eigenvalues, for each desired value of h, where

$$D(h) = \left(\begin{array}{cc|c} G(h) & Q(h)\ (A_2 + D_2) & \\ h(A_3 + D_3) & I_{M-K} + h\ D_4 & I_M \\ \hline Q(h)\ D_1 & \Theta_1 & \\ \Theta_2 & h\ A_4 & \Theta_M \end{array}\right)$$

Some of the procedures in steps (1) to (4) above can be carried out using standard library subroutines. Further, the process can be repeated to cover a range of values of step-length, h.

3.2.1 Two approaches for step (2), and merits of each

Initially, the direct method for calculating the fundamental matrix Φ of A_1, was chosen as being superior, because it gives an exact solution, whereas the numerical integration technique gives an approximate solution. However, it was necessary to cater for both approaches in the software package, since the properties of the eigenvalues of A_1 may restrict the use of one or other method. A method of analysis was derived and is described in Appendix II, which allows a choice of step-length for the numerical integration, based on stability and accuracy of the solution. In fact, the results for both methods for obtaining $\Phi(t)$ did not differ, for the purpose of determining stability.

It appears that neither method has any particular merit over the other, except when the properties of the eigenvalues of A_1 restrict the use of either method.

It is suggested that the numerical integration method be used as a first approach. Regardless of the choice of method for computing $\Phi(t)$, the software package calculates the eigenvalues of A_1.

3.2.2 Further comments on step 3

It is possible to devise a partitioning scheme in which the analogue portion is unstable. This form of partitioning is considered undesirable and the software package does address this question. In step (3) the requirement states that G(h), and its stability, be examined. G(h) corresponds to the analogue part of the simulation. This reduces to determining that the eigenvalues of G should not be greater than unity in modulus. This will be true if and only if there are no eigenvalues of A_1 with positive real part. This constraint is incorporated into the software package output to inform the user. However, the package will continue to determine the stability of the overall simulation using the direct method to compute $\Phi(h)$, but the user is reminded that such stability, if obtained, will apply only for the linearized system. This is considered to be the best the software package can do in the circumstances. The problems inherent in simulating an unstable system on the analogue portion, e.g. limiting because of particular choices of scaling, are clearly problems which must be left to the user.

4. DESCRIPTION OF THE SOFTWARE PACKAGE AND ITS USAGE

The purpose of this Section is to provide, with reasonable completeness, the information required by a potential user of the package.

4.1 Description of the software package

The package is programmed in Fortran for an IBM 370/168 digital computer. It uses library routines from the Harwell library, for finding the inverse of a complex matrix, and the I.M.S.L. library for computing the eigenvalues and associated eigenvectors of both real and complex matrices.

The program is best described with the aid of a flowchart, shown in figure 2. The matrices A_1, ... D_4 and their dimensions are first defined and printed. The eigenvalues and eigenvectors of A_1 are computed, and the properties of the real part of the eigenvalues are examined. The eigenvalues are also examined to establish if any are repeated. Next, the fundamental matrix for A_1, and the matrices G and Q are computed for a range of step-lengths, h, chosen by the user. The direct or numerical integration, (N.I), method is used for the previous step, depending on user choice, or whether the eigenvalues of A_1 are distinct, (N.I. method), or the real part of any eigenvalue is positive (direct method). For each step-length, h, the partitioned matrix D is computed, and its eigenvalues are examined to determine the stability of D. The main program prints information at intervals throughout each stage of program execution.

The Fortran coding and a description of the variables used are presented in Appendix III and IV. The program uses double precision arithmetic The matrices A_1, ... A_4, D_1, ... D_4, G(h), Q(h) and D(h) are real, while the eigenvalues and fundamental matrix for A_1 are, in general, complex. As a result, some of the subroutines are more involved than might otherwise be expected.

4.1.1 Main program

The main program accepts the input data, controls the main branches of the analysis, namely which method is used to calculate Φ, G(h), and Q(h), and prints the results. It makes calls directly to subroutines EIGEN, FUND, LAMBDA, NUMERI, DMATRX, MODEIG, and OUTPUT, which are described below.

4.1.2 Subroutine EIGEN

This routine controls the computation of the eigenvalues, λ_i, and the matrix of eigenvectors, C, for A_l, via subroutine CNVRT. ($C = P^{-1}$ in the theory of reference 1, Appendix I). EIGEN calls subroutine DIST, after which it tests if the numerical integration technique has been chosen by the user, and whether λ_i are repeated. If the latter situation exists, the routine forces the program to use the N.I. method. If neither of the above criteria is true the routine computes the inverse, B, of the matrix of eigenvectors, C, via subroutine CNVRT, in preparation for computing the fundamental matrix corresponding to A_l by the eigenvalue method. Should the inverse routine fail, the N.I. method is chosen.

4.1.3 Subroutine DIST

Specifies the real parts of each of the λ_i, for determining the stability of G(h), and also establishes whether the λ_i are repeated, i.e., whether $\lambda_i = \lambda_j$ for $i \neq j$.

4.1.4 Subroutine FUND

The purpose of this routine is to define the fundamental matrix, $\Phi(h)$, by direct calculation, using eigenvalue theory. For each value of h, the routine computes two complex diagonal matrices diag $(\exp \lambda_1 h, \ldots \exp \lambda_k h)$ and diag $(\exp \frac{\lambda_1 \ h-1}{\lambda_1}, \cdots \exp \frac{\lambda_k \ h-1}{\lambda_k})$, using a complex matrix multiplication routine, DCMPY. It is expected that the imaginary elements of the above matrices will be less than 10^{-3} in magnitude. The matrices are redefined as G(h) and Q(h). A diagnostic is printed if the imaginary elements of the complex matrices do not conform to the above requirement, but the program continues to execute.

4.1.5 Subroutine NUMERI

This routine provides the control for the numerical integration routines. The numerical integration step-length is established by routines LAMBDA called from MAIN, and STEP, and the integration, using the "predictor-corrector" method, is performed in subroutine SINT. NUMERI increments time up to the required value of h, and computes G or Q depending on the value assigned to a variable NAM.

4.1.6 Subroutine LAMBDA

In this subroutine the analysis of Appendix II is applied to establish the numerical integration step-length, chosen for stability and accuracy of the solution. Values for the N.I. step-length, h, are computed as a function of the real part of each eigenvalue, λ_i, and must also meet a criteria related to the imaginary part of the eigenvalue λ_i (as described in Appendix II).

If this criteria is met for all the eigenvalues, the smallest h is chosen for the N.I. step-length.

If the above criteria is not met for one or more of the eigenvalues, then h_l is chosen for the respective eigenvalues according to the analysis of Appendix II, and the N.I. step-length is the minimum over all the eigenvalues of the h and h_l's respectively. h must also be at least as small as the minimum interval between the step-lengths chosen by the software package user for examining the stability of the hybrid solution.

The subroutine is entered from the MAIN program, and provides the N.I. step-length, h_{STEP}, for further refining in subroutine STEP.

4.1.7 Subroutine STEP

Subroutine LAMBDA defines the N.I. step-length purely on the basis of an accuracy criteria, and the largest eigenvalue. It is necessary to establish individual N.I. step-lengths for each interval of stability step-lengths chosen by the user. The numerical integration step-length, h_{STEP}, for the interval (h_i, h_{i+1}) is modified, such that

$$h_S = h_{STEP} \pm \epsilon$$

where ϵ is a small quantity, and there are exactly an integral number of integration steps in the interval (h_i, h_{i+1}). Subroutine STEP is called from subroutine NUMERI whenever a new N.I. step-length is required. The subroutine returns the integration step-length h_S.

4.1.8 Subroutine SINT

The predictor-corrector steps for numerical integration are defined in subroutine SINT. The calling vector defines IPASS and NAM, to control which branch of the routine is to be executed, i.e. the "predict" or "correct" step, and for identifying which of functions G or Q is being computed.

4.1.9 Subroutine DMATRX

The partitioned 2M x 2M matrix D is defined in this subroutine. It calls the subroutine DMMPY, to perform multiplication of two real matrices.

4.1.10 Subroutine MODEIG

The routine controls CNVRT to compute the eigenvalues μ_i, of D, and then tests that

$$|\mu_i| \leq 1, \; i = 1, \ldots, 2M$$

If this criteria does not hold for all μ_i, the routine records which of the eigenvalues are in modulus too large, and their respective values, for printing by the main program.

4.1.11 Subroutine CNVRT

This subroutine controls all calls to the library routines. It has three branches, each being defined by a value I, assigned in the calling subprogram according to whether eigenvalues and eigenvectors of a matrix are required (I = 1), the inverse of a matrix is required (I = 2), or eigenvalues only are required (I = 3). EIGEN calls CNVRT for I = 1, 2, and MODEIG calls CNVRT for I = 3.

For I = 1, CNVRT calls EIGRF from the IMSL library, which computes complex eigenvalues and complex eigenvectors of A_1, and a performance index for the mathematical accuracy of the technique. EIGRF returns the performance index and CNVRT prints a diagnostic. It is possible that if the eigenvalues are not distinct, then the matrix of eigenvectors is incorrect. The software package prevents this matrix being used further, by forcing the program to use the numerical integration method for computing the fundamental matrix $\Phi(h)$, whenever the eigenvalues are repeated. The eigenvalues are not normalised by EIGRF.

For I = 2, CNVRT computes the inverse matrix B of the complex matrix of eigenvectors C, using the function MA23BD of the Harwell library routine MA23AD. The library routine overwrites the input matrix, and returns the inverse matrix in the storage provided by it. It is therefore necessary for the program to remember C before calling MA23BD.

The routine MA23BD makes no allowance for the size of the matrix in the dimension or common statement of the calling subprogram to be bigger than the matrix row dimension in the calling statement. Therefore it is necessary to define another complex matrix, Y, having a variable dimension in the calling subprogram CNVRT. This branch of CNVRT is only executed with IMETH = 1.

For I = 3 CNVRT calls EIGRF, to compute the complex eigenvalues of the matrix D. This call to CNVRT is made from subroutine MODEIG

4.1.12 Subroutine DMMPY and DCMPY

Subroutine DMMPY performs matrix multiplication of two real matrices. The matrices need not be square, but the program tests that the column dimension of the first matrix equals the row dimension of the second matrix. The output matrix is 10 x 10, even though the submatrix containing the solution is MMM x LLL which may be smaller than 10 x 10. DCMPY is the complex version of DMMPY.

4.1.13 Subroutine OUTPUT

This subroutine controls the printing of the input matrices A_1, D_1, ... D_4.

4.1.14 Library routines

EIGRF and MA23BD are well documented in the Computing Services Group Library of W.R.E. The Harwell and IMSL libraries are both extensively used and tested by other computing facilities, and have performed well in this software package.

4.2 Software package running

This Section covers the data requirements, the program output, and the method of creating a data file under the IBM 370/168 supplied TSO system. Operational characteristics are explained, and the identification of the W.R.E. Library version of the software package is supplied.

4.2.1 Data input and program output

Table 1 lists the input parameters and the form in which they must be presented, but further discussion of the parameters is presented below.

Matrices A1, D1, ... A4 must be defined in terms of their sizes and elements. In most system problems, some at least, of these matrices will be null, and the others will very likely be sparse. The user need only define the non-zero elements, and the respective row and column of each element, for the non-null matrices.

TABLE 1. DATA REQUIREMENTS FOR SOFTWARE PACKAGE

Data record	Parameter value	Parameter description	Constraints on parameters
1	K,M	K = Row size of A1 matrix M = total number of integrals computed analogue or digitally	$K \leq 10$ (nominally) $K + 1 \leq M \leq 20$ (nominally) If no integrals are assigned to digital computer i.e. M = K, then M set to K + 1
2	NEL	Numbers of non-zero elements in A1	
3	I,J,A1(I,J),... for NEL elements	I = row number J = column number A1(I,J) = non zero element of A1 in I^{th} row and J^{th} column.	I,J are integers A1(I,J) is a floating point value. These values are separated by commas.
4	NEL	Number of non-zero elements for D1	
.			
.			
.			

. This set continues until A1, D1, ..., A4, D4 have all been defined. If NEL is zero for any of these matrices, the next data record contains NEL for the next matrix in the set. The order of input of the matrices must run A1, D1, A2, D2, A3, D3, A4, D4. The data, if all these matrices have non-zero elements, are defined in records 2 to 17. For most problems there will be less records, but NEL must be defined for each matrix, even if it is assigned the value zero.

18 (or less)	NT,EL,HINT	NT = number of user defined step-lengths EL = value of 1st required step-length HINT = time interval between successive step-lengths.	The values for EL and HINT, are required if the step-lengths are at equal intervals. Otherwise EL = 0, HINT = 0. If only one step-length is to be defined HINT = 0.
19 (or less)	IMETH	User option for choosing direct method for computing $\Phi(h)$ i.e. IMETH = 1, or N.I. method i.e. IMETH = 2.	IMETH = 1 may be overwritten in program for N.I. method. IMETH = 2 is considered to be accurate enough, in general, and is recommended to user.
20 (or less)	H(I),I=1..,NT	H(I) are the user defined step-lengths. NT was defined earlier.	This record is optional. It will not be needed if H(I) is fully defined by data record 18.

The matrices are ordered implicitly A1, D1, A2, D2, A3, D3, A4, D4 in the data.

A variable NEL defines the number of non-zero elements for each matrix in turn. If NEL is zero, then the matrix is assumed null, and NEL for the next matrix of the set is the next data required. When NEL is non-zero, the next data input defines all the non-zero elements (NEL of them) for the matrix in question.

Consider the example, in which there are 4 non-zero elements

$$A1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Assuming the matrix dimension, i.e. 3 x 3, has already been defined, then two data records are required to fully define A1. The first data record contains the value of NEL. The second data record contains the row, column and value of each non-zero element in that strict order, although the order in which each element group is defined is unimportant.

The row and column are defined by integers, and the element value is defined by a floating point constant. The two data records defining A1 are:

```
4
1,1,1.0,2,2,2.0,3,2,1.0,3,3,1.
```

Following the matrix definition for A1, ..., D4, the user must define a value, or values, of step-length h, for which he wishes to test his partitioning scheme. The program accepts variables to define a range of values, h, at equal intervals, or a series of values h, or will read a single value for h. In the last instance, the software package provides the stability for the single value, h, and also for 1.5 times that value. This may give the user a better idea of where his choice lies relative to stability. If the step-lengths are at equal intervals, the package computes the array H(I), from the value of H(1), and the time interval between step-lengths, up to NT values.

The user has a choice as to which method is used to compute G(h) and Q(h) via the fundamental matrix $\Phi(h)$. The direct calculation, using eigenvalue theory, is allocated by the variable IMETH = 1. The approximate solution, using numerical integration, is assigned in the package by IMETH = 2. Results from either method have not differed significantly for the purpose of stability analysis, and the user is advised to use the numerical integration method, if he has no preference.

Initially, the program print-out defines the input matrices A1, D1, ..., D4, together with the values of h for program execution. The output then defines a performance index for the eigenvalue routine, followed by the eigenvalues of A1, and an error parameter indicating success (IER = 0) or failure (IER > 0). The distinctness, or otherwise, of the eigenvalues is defined. A comment is printed concerning the stability of the matrix G, which provides the user a very important guide to the likely success or failure of his partitioning scheme. This stability indication is independent of values of h. Next, the method, by which the matrices G and Q will be computed, is printed. The preceding information is printed once for each run.

The next block of output is repeated for each value of h ascribed by the user. The step-length, h, in seconds, the N.I. step-length and accuracy, if applicable, and the matrices, G, Q and D are printed. The value of the error parameter for computing the eigenvalues of D is given and the eigenvalues are printed. Finally, there is a statement concerning the stability of the scheme for the present value of h.

4.2.2 Creating a data file

In order to run the software package under TSO, or via a batch job, the user must supply a data file, and link this file with the Fortran program.

For most user problems, the matrices A1, D1 may be of the order 6 x 6, or bigger. The record length required for specifying the non-zero elements of such matrices will very likely exceed 80 characters. This is the default value assigned in the EDIT command or by the JCL accompanying the data cards, to the record length parameter LRECL (or LINE). (The 8 characters used to specify the line number are included in the 80). A record length of 120 characters is considered adequate for problems likely to be encountered, but in any case the value must not exceed 255.

The BLKSIZE (or BLOCK) parameter of the EDIT command specifies the maximum length (in bytes) for all the records of the data set. Hence the BLKSIZE parameter must be an integer multiple of the LRECL parameter. The default parameter is 3120, (also the maximum allowed), which conveniently is a multiple of 120, but, in any case, a smaller number may be specified. For the test data sets, described in this report, the BLKSIZE parameter has been specified as 2640, allowing for a maximum of 22 records of 120 characters. The BLKSIZE and LRECL parameters are only specified when a new data set is created.

The EDIT command for creating the tracking radar simulation data set, SEEK4, of Section 5 is given as an example.

```
EDIT  SEEK4  DATA  NEW  BLOCK(2640)  LINE(120)
```

or

```
EDIT  SEEK4  DATA  NEW  BLKSIZE(2640)  LRECL(120)
```

4.2.3 Operational characteristics

This Section covers the general operating characteristics of interest to a user.

The W.R.E. Library program identification for the software package is 4154, and a user will be supplied with the data-set name of the package, and any other JCL routines necessary for running the program, on request from Computing Services Group.

(a) Running time

The program typically compiles and executes, for up to twelve values of step-length h, in under ten seconds. Execution time for the same run is less than three seconds.

(b) Diagnostics

The program terminates in the event that neither the N.I., nor the direct method for computing Φ, can be applied. This will occur if some of the eigenvalues are repeated, and one or more have positive real part.

Most of the printing is controlled by the MAIN program.

However, there are several instances where other subroutines may print diagnostics. In sub-routine FUND, if the complex versions of the matrices G and Q have imaginary elements larger in magnitude than 10^{-3}, a diagnostic is printed. The program continues executing.

In subroutines DMMPY and DCMPY, if the matrices supplied are not compatible for multiplication, then a diagnostic is printed, and the program terminates.

Whenever subroutine CNVRT uses the IMSL routine EIGRF to compute eigenvalues and eigenvectors of A_1, a performance index is computed defining how the IMSL routines performed. CNVRT prints the appropriate diagnostic, and the package continues to execute. EIGRF also provides an error parameter, IER, and its value is printed in CNVRT. IER may take the following range of values,

IER = 0	Routines worked normally.
IER $\geq$ 128	The numerical solution failed to converge for one of the eigenvalues.
IER $<$ 128	One of the parameters in the calling statement for EIGRF is inconsistent.

The user is advised to regard the program results as incorrect whenever the error parameter is non-zero. It may then be necessary for the user to refer to the write-up of the IMSL routine EIGRF in order to satisfactorily continue with the problem solution.

(c) Replacing software package routines

Should a user need to modify or replace any of the routines of the software package, the point is made that modifications or new routines must use double precision arithmetic.

5. APPLICATIONS

Any physical system if analysed in great detail is non-linear. Most physical systems are also time-varying, but if changes in the system characteristics are very slow, compared with variations in the input, a linear time-varying system can be approximated by a linear time-invariant one.

It has been assumed here that a potential software package user has derived a linearized version for the system, prior to considering a hybrid simulation model, and that these equations are in the form

$$\dot{\underline{x}} = A\underline{x}$$

as required by the software package.

However, it sometimes occurs, that in neglecting small time constants in the analogue portion of the model certain of the system equations are algebraic equations rather than differential equations. This is not covered by the theory presented here. Rather than extend the theory to cover the general case where equations may be algebraic or differential, it is considered easier to reintroduce small time constants for the purpose of stability analysis.

The following sections deal with two linear systems (see reference 3 for a detailed derivation of the linearization process). In fact, the existing digital simulation model for example 2, represents the control fin actuator, for a missile, as responding instantaneously, to produce a fin deflection, η, to the autopilot error signal, $\bar{\eta}$. A lag of 0.001 s is introduced to express the system in a form suitable for software package implementation.

5.1 Tracking radar simulation

The first example describes a (single plane) simulation of a radar guided weapon homing on a target (figure 3), in which the motion of the missile (under autopilot control) is simulated on the analogue computer, while the signal processing functions of the missile radar receiver and the dynamics of the radar servo and pedestal systems are represented on the digital computer. The configuration is as shown in figure 4.

The linearized equations which describe the system are

$$\psi_{DM} = F(S)\ (\psi_I - \psi_M) \tag{2}$$

$$\gamma = -3(\psi_{DM} + \psi_M) \tag{3}$$

$$\dot{\psi}_I = \gamma \tag{4}$$

$$\dot{\psi}_M = r \tag{5}$$

and

$$r = G(S)\ \gamma \tag{6}$$

where

ψ_{DM} = dish-body angle

ψ_M = missile azimuth

γ = required dish rate to maintain track

r = missile angular velocity

$$F(S) = \frac{(1 + T_{D1}\ S)}{1 + T_{D2}\ S + T_{D3}\ S^2}$$

and

$$G(S) = \frac{K(1 + T_{A1}\ S)(1 + T_{A2}\ S)}{(1 + T_{A3}\ S)(1 + T_{A4}\ S + T_{A5}\ S^2)}$$

The partitioning scheme is such that ψ_{DM} and γ, and hence ψ_I, are evaluated and integrated digitally, whilst ψ_M and r are evaluated and integrated on the analogue computer.

It is necessary to reduce the second order equation

$$\psi_{DM} = F(S)\ (\psi_I - \psi_M)$$

to two first-order equations. Expanding (2), we have

$$\psi_{DM} + T_{D2}\,\dot{\psi}_{DM} + T_{D3}\,\ddot{\psi}_{DM} = \psi_I - \psi_M + T_{D1}\,\dot{\psi}_I - T_{D1}\,\dot{\psi}_M \tag{7}$$

Define

$$\dot{Z}_1 = \psi_{DM} - \psi_I + \psi_M \tag{8}$$

Then equation (7) reduces to

$$\ddot{\psi}_{DM} = \frac{-T_{D2}}{T_{D3}}\,\dot{\psi}_{DM} - \frac{1}{T_{D3}}\,\dot{Z}_1 + \frac{T_{D1}}{T_{D3}}\,\dot{\psi}_I - \frac{T_{D1}}{T_{D3}}\,\dot{\psi}_M$$

Integrating the above equation, we have

$$\dot{\psi}_{DM} = \frac{-T_{D2}}{T_{D3}}\,\psi_{DM} - \frac{1}{T_{D3}}\,Z_1 + \frac{T_{D1}}{T_{D3}}\,\psi_I - \frac{T_{D1}}{T_{D3}}\,\psi_M$$

which, together with equation (8) define equation (2) as two first order d.e.'s. We see that $(\dot{\psi}_{DM})_{N+1}$ which is calculated in the interval $t_N < t < t_{N+1}$ on the digital computer is a function of $(\psi_{DM})_N$ and $(\psi_I)_N$ from the digital, and ψ_M from the analogue computer in the interval $t_{N-1} < t < t_N$.

It is also necessary to reduce the third order equation

$$r = G(S)\,\gamma$$

to three first order d.e.'s.

Express equation (6) as

$$r = \frac{K(1 + T_{A1}\,S)\,\gamma}{1 + T_{A3}\,S}\left[\frac{1 + T_{A2}\,S}{1 + T_{A4}\,S + T_{A5}\,S^2}\right] \tag{9}$$

and define

$$u = \frac{K(1 + T_{A1}\,S)}{1 + T_{A3}\,S} \tag{10}$$

We can then define

$$Z_2 = \frac{K\,\gamma}{1 + T_{A3}\,S}$$

and hence

$$\dot{Z}_2 = \frac{K S \gamma}{1 + T_{A3} S}$$

$$= \frac{K \gamma - Z_2}{T_{A3}}$$

$$= \frac{K}{T_{A3}} \gamma - \frac{1}{T_{A3}} Z_2$$

Therefore equation (10) becomes

$$u = Z_2 + T_{A1} \dot{Z}_2$$

Now for equation (9) we have a new expression

$$r = \frac{1 + T_{A2} S}{1 + T_{A4} S + T_{A5} S^2} u$$

$$r + T_{A4} \dot{r} + T_{A5} \ddot{r} = u + T_{A2} \dot{u}$$

If we put

$$\dot{Z}_3 = r - u$$

and substitute for $\dot{Z}_3$ in the above equation, we obtain

$$\ddot{r} = \frac{-T_{A4}}{T_{A5}} \dot{r} - \frac{1}{T_{A5}} \dot{Z}_3 + \frac{T_{A2}}{T_{A5}} \dot{u}$$

Integrating, gives

$$\dot{r} = \frac{-T_{A4}}{T_{A5}} r - \frac{1}{T_{A5}} Z_3 + \frac{T_{A2}}{T_{A5}} u$$

Expressing u in terms of Z_2 and $\dot{Z}_2$, we derive

$$\dot{r} = \frac{-T_{A4}}{T_{A5}} r - \frac{1}{T_{A5}} Z_3 + \frac{T_{A2}}{T_{A5}} \left[Z_2 + T_{A1} \left(\frac{K}{T_{A3}} \gamma - \frac{1}{T_{A3}} Z_2 \right) \right]$$

$$= \frac{-T_{A4}}{T_{A5}} r - \frac{1}{T_{A5}} Z_3 + \frac{T_{A2}}{T_{A5}} \left[1 - \frac{T_{A1}}{T_{A3}} \right] Z_2 + \frac{K T_{A1} T_{A2}}{T_{A3} T_{A5}} \gamma$$

Equation (6) has been reduced to three first order d.e.'s, $\dot{r}$, $\dot{Z}_2$ and $\dot{Z}_3$. We shall assume that $\dot{Z}_2$ and $\dot{Z}_3$ are computed on the analogue computer, and can therefore see that the element γ is computed digitally in the interval $t_{N-1} < t < t_N$, and passed to the analogue portion via the D/A interface at time t_N, for computation of $\dot{r}$, $\dot{Z}_2$ and $\dot{Z}_3$ in the interval $t_N < t < t_{N+1}$.

Equations (2) to (6) can now be expressed by a set of equations defined in (11).

$$
\begin{aligned}
\dot{\psi}_M &= r \\
\dot{r} &= \frac{-T_{A4}}{T_{A5}} r + \frac{T_{A2}}{T_{A5}}\left[1 - \frac{T_{A1}}{T_{A3}}\right] Z_2 - \frac{1}{T_{A5}} Z_3 + \frac{K\, T_{A1}\, T_{A2}}{T_{A3}\, T_{A5}} [\gamma]_N \\
\dot{Z}_2 &= - \frac{1}{T_{A3}} Z_2 + \frac{K}{T_{A3}} [\gamma]_N \\
\dot{Z}_3 &= r - \left[1 - \frac{T_{A1}}{T_{A3}}\right] Z_2 - \frac{K\, T_{A1}}{T_{A3}} [\gamma]_N \\
\dot{\psi}_{DM} &= - \frac{T_{D1}}{T_{D3}} \psi_M - \frac{T_{D2}}{T_{D3}} \psi_{DM} + \frac{T_{D1}}{T_{D3}} \psi_I - \frac{1}{T_{D3}} Z_1 \\
\dot{\psi}_I &= [\gamma]_{N+1} \\
\dot{Z}_1 &= \psi_M + \psi_{DM} - \psi_I
\end{aligned}
\tag{11}
$$

where

$$
[\gamma]_N = -3[\psi_{DM}]_N - 3[\psi_M]_{N-1}
$$

Expressing equations (11) as d.e.'s which are calculated in the interval $t_N < t < t_{N+1}$ by the hybrid computer, we have

$$\dot{\psi}_M = r$$

$$\dot{r} = \frac{-3K\,T_{A1}\,T_{A2}}{T_{A3}\,T_{A5}}[\psi_M]_{N-1} - \frac{T_{A4}}{T_{A5}}r + \frac{T_{A2}}{T_{A5}}\left[1 - \frac{T_{A1}}{T_{A3}}\right]Z_2 - \frac{1}{T_{A5}}Z_3 - \frac{3K\,T_{A1}\,T_{A2}}{T_{A3}\,T_{A5}}[\psi_{DM}]_N$$

$$\dot{Z}_2 = \frac{-3K}{T_{A3}}[\psi_M]_{N-1} - \frac{1}{T_{A3}}Z_2 - \frac{3K}{T_{A3}}[\psi_{DM}]_N$$

$$\dot{Z}_3 = \frac{3K\,T_{A1}}{T_{A3}}[\psi_M]_{N-1} + r - \left[1 - \frac{T_{A1}}{T_{A3}}\right]Z_2 + \frac{3K\,T_{A1}}{T_{A3}}[\psi_{DM}]_N$$

$$\dot{\psi}_{DM} = \frac{-T_{D1}}{T_{D3}}[\psi_M]_N - \frac{T_{D2}}{T_{D3}}\psi_{DM} + \frac{T_{D1}}{T_{D3}}\psi_I - \frac{1}{T_{D3}}Z_1$$

$$\dot{\psi}_I = -3[\psi_M]_N - 3\,\psi_{DM}$$

$$\dot{Z}_1 = [\psi_M]_N + \psi_{DM} - \psi_I$$

(12)

The analogue vector χ, and the digital vector z are defined as

$$\chi = \begin{pmatrix} \psi_M \\ r \\ Z_2 \\ Z_3 \end{pmatrix}, \text{ and } z = \begin{pmatrix} \psi_{DM} \\ \psi_I \\ Z_1 \end{pmatrix}.$$

It remains to express the set of equations defined in (12) in the form defined in Appendix I.

$$\dot{\chi} = \begin{pmatrix} \dot{\psi}_M \\ \dot{r} \\ \dot{Z}_2 \\ \dot{Z}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-T_{A4}}{T_{A5}} & \frac{T_{A2}}{T_{A5}}\left(1 - \frac{T_{A1}}{T_{A3}}\right) & -\frac{1}{T_{A5}} \\ 0 & 0 & -\frac{1}{T_{A3}} & 0 \\ 0 & 1 & -\left(1 - \frac{T_{A1}}{T_{A3}}\right) & 0 \end{pmatrix} \begin{pmatrix} \psi_M \\ r \\ Z_2 \\ Z_3 \end{pmatrix}$$

$$+ \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{-3K\ T_{A1}\ T_{A2}}{T_{A3}\ T_{A5}} & 0 & 0 & 0 \\ \frac{-3K}{T_{A3}} & 0 & 0 & 0 \\ 3\ \frac{K\ T_{A1}}{T_{A3}} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi_M \\ r \\ Z_2 \\ Z_3 \end{pmatrix}_{N-1}$$

$$+ \begin{pmatrix} 0 & 0 & 0 \\ \frac{-3K\ T_{A1}\ T_{A2}}{T_{A3}\ T_{A5}} & 0 & 0 \\ \frac{-3K}{T_{A3}} & 0 & 0 \\ \frac{3K\ T_{A1}}{T_{A3}} & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi_{DM} \\ \psi_I \\ Z_1 \end{pmatrix}_N \qquad (13)$$

i.e.

$$\dot{\underset{\sim}{x}} = A_1 \underset{\sim}{x} + D_1 \underset{\sim}{x}_{N-1} + D_2 \underset{\sim}{z}_N$$

and A_2 a null matrix. The equation (13) defines the analogue portion of the partitioning scheme.

The non-zero elements of the matrix D_2 are components of $\dot{r}$, $\dot{Z}_2$ and $\dot{Z}_3$, and are computed on the digital computer as shown in figure 4. It is immaterial whether the elements of D_2 are computed digitally or on the analogue portion (then called A_2) since the two matrices occur as a sum $(A_2 + D_2)$ in the matrix, D, of the theory. Similarly, it is immaterial which of the matrices A_3 or D_3 is defined in the expression for $\dot{z}$.

Further, expressing (12) in the form of Appendix I for $\dot{z}$.

$$\dot{\underset{\sim}{z}} = \begin{pmatrix} \frac{-T_{D1}}{T_{D3}} & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi_M \\ r \\ Z_2 \\ Z_3 \end{pmatrix}_N + \begin{pmatrix} \frac{-T_{D2}}{T_{D3}} & \frac{T_{D1}}{T_{D3}} & \frac{-1}{T_{D3}} \\ -3 & 0 & 0 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} \psi_{DM} \\ \psi_I \\ Z_1 \end{pmatrix}_N \qquad (14)$$

$$= D_3 \underset{\sim}{x}_N + D_4 \underset{\sim}{z}_N \text{ (or } A_3 \underset{\sim}{x}_N + D_4 \underset{\sim}{z}_N)$$

with A_3, A_4 null matrices.

Consider as an example under consideration the situation where

$$T_{D1} = 0.1$$

$$T_{D2} = 0.22$$

$$T_{D3} = 0.016$$

$$K = 3.52$$

$$T_{A1} = 0.01$$

$$T_{A2} = 1.9$$

$$T_{A3} = 0.023$$

$$T_{A4} = 0.54$$

$$T_{A5} = 0.094$$

Then the data for defining the configuration for the software package is

$$K = 4$$

$$M = 7$$

$$M-K = 3$$

A_1, D_1 are 4 x 4 matrices

A_2, D_2 are 4 x 3 matrices

A_3, D_3 are 3 x 4 matrices

and

A_4, D_4 are 3 x 3 matrices

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -5.8 & 11.4 & -10.6 \\ 0 & 0 & -43.5 & 0 \\ 0 & 1 & -0.6 & 0 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -92.8 & 0 & 0 & 0 \\ -459.1 & 0 & 0 & 0 \\ 4.6 & 0 & 0 & 0 \end{pmatrix}$$

$$A_2 = \theta \qquad D_2 = \begin{pmatrix} 0 & 0 & 0 \\ -92.8 & 0 & 0 \\ -459.1 & 0 & 0 \\ 4.6 & 0 & 0 \end{pmatrix}$$

$$A_3 = \theta \qquad D_3 = \begin{pmatrix} -6.3 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$A_4 = \theta \qquad D_4 = \begin{pmatrix} -13.8 & 6.3 & -62.5 \\ -3 & 0 & 0 \\ 1 & -1 & 0 \end{pmatrix}$$

Table 2 presents the data in the form accepted by the package, and Appendix VI shows a sample output from the computer run.

TABLE 2. INPUT DATA FOR EXAMPLE 1

Parameter name	Data Record
K,M	4,7
NEL	7
A1	1,2,1.0,2,2,-5.8,4,2,1.0,2,3,11.4,3,3,-43.5, 4,3,-0.6,2,4,-10.6
NEL	3
D1	2,1,-92.8,3,1,-459.1,4,1,4.6
NEL (A2)	0
NEL	3
D2	2,1,-92.8,3,1,-459.1,4,1,4.6
NEL (A3)	0
NEL	3
D3	1,1,-6.3,2,1,-3.0,3,1,1.0
NEL (A4)	0
NEL	6
D4	1,1,-13.8,2,1,-3.0,3,1,1.0,1,2,6.3,3,2, -1.0,1,3,-62.5
NT, EL, HINT	6,0.038,0.001
IMETH	2

5.2 Height control system simulation

The second example concerns the simulation of the height control system of a missile. For the present purposes, the system is assumed to have the configuration shown in figure 5. In this, the indicated missile height and velocity are obtained by integration of the missile acceleration, as sensed by the inertial platform. Drifts and biases are corrected by comparison with the indicated height and the smoothed output of the missile's altimeter. The indicated height, Z, is then compared with the demanded height, Z_D, and, with the vertical velocity, v, is used to form the demanded acceleration, a_D. This is the input to the missile autopilot which controls the missile aerodynamic response such that the achieved vertical acceleration, a_Z, closely follows the demanded input.

One approach to the hybrid solution is to simulate the autopilot and inertial platform and a portion of the aerodynamics on the analogue computer. Evaluation of the missile acceleration and the derivative of pitch rate are computed digitally, since they are functions of several variables, and integrated on the analogue portion (figure 6). The above configuration is the "text-book" version of the partitioning problem viz, that algebra should be solved on the analogue computer.

The analysis of reference 1 has shown the potential danger of applying this "text-book" approach, if the resulting scheme evaluates derivatives on one portion of the computer and integrates them on the other. This "complete partitioning" can result in an even less stable division than evaluating and integrating derivatives entirely on the digital computer, user Euler integration. Nevertheless, without the guidance which the analysis provides,

this partitioning scheme might well appear to be a sensible one, and, as such, provides an interesting test case for the theory. The linearized system equations are defined below.

$$\dot{Z} = v$$

$$\dot{v} = a_Z$$

$$a_Z = a_1 \alpha + a_2 \eta$$

$$\dot{q} = a_3 \alpha + a_4 \eta$$

$$\dot{\alpha} = \frac{a_Z}{u} + q$$

$$\dot{\bar{\eta}} = \frac{K_{A4}}{\tau}\left[\frac{K_{A1}(1 + T_{A1} S)}{(1 + T_{A2} S)(1 + T_{A3} S)} q - K_{A5} a_Z + a_D\right]$$

$$\eta = \frac{1}{1 + \tau S}\bar{\eta}$$

$$a_D = K_{A2}(Z_D - Z) - K_{A3} v$$

For stability analysis we can regard Z_D as zero. Thus

$$a_D = -K_{Z2} Z - K_{A3} v$$

The closed loop response for missile height and acceleration are,

$$Z = \frac{Z_D}{(1 + 0.85 S)(1 + 0.15 S + 0.039 S^2)}$$

and

$$a_Z = \frac{3(1 - 0.0035 S)}{(1 + 0.18 S)(1 + 0.016 S)} a_D$$

In the above equations a_1, a_2, a_3 and a_4 are functions of several variables related to missile incidence, Mach number, and air pressure.

a_D = demanded missile acceleration

Z_D = demanded missile height

u = missile velocity

$K_{A1}, \ldots K_{A5}$ = gains in pitch autopilot

T_{A1}, T_{A2}, T_{A3} = time constants in body rate feedback of autopilot

Thus the linearized equations for stability analysis are

$$\dot{z} = v \tag{15}$$

$$\dot{v} = a_1 \alpha + a_2 \eta \tag{16}$$

$$\dot{q} = a_3 \alpha + a_4 \eta \tag{17}$$

$$\dot{\alpha} = \frac{1}{u}(a_1 \alpha + a_2 \eta) + q \tag{18}$$

$$\dot{\eta} = \frac{K_{A4}}{\tau}\left[\frac{K_{A1}(1 + T_{A1} S)}{(1 + T_{A2} S)(1 + T_{A3} S)} q - K_{A5}(a_1 \alpha + a_2 \eta) - K_{A2} z - K_{A3} v\right] - \frac{\eta}{\tau} \tag{19}$$

We must reduce the third order d.e. defined by equation (19) to three first order d.e.'s.

Put

$$\beta = \frac{K_{A1}}{1 + T_{A2} S} q$$

then

$$\dot{\beta} = \frac{K_{A1}}{T_{A2}} q - \frac{1}{T_{A2}} \beta$$

We will assume that $\dot{\beta}$ is computed on the analogue computer. Equation (19) now becomes

$$\dot{\eta} = \frac{K_{A4}}{\tau}\left[\frac{(1 + T_{A1} S)}{(1 + T_{A3} S)} \beta - K_{A5}(a_1 \alpha + a_2 \eta) - K_{A2} z - K_{A3} v\right] - \frac{\eta}{\tau} \tag{20}$$

Put

$$\delta = \frac{(1 + T_{A1} S)}{(1 + T_{A3} S)} \beta$$

Thus

$$\dot{\delta} = \frac{-\delta}{T_{A3}} + \frac{\beta}{T_{A3}} + \frac{T_{A1}}{T_{A3}} \dot{\beta}$$

$$= \frac{-\delta}{T_{A3}} + \frac{T_{A1} K_{A1}}{T_{A3} T_{A2}} q + \frac{\beta}{T_{A3}}\left[1 - \frac{T_{A1}}{T_{A2}}\right]$$

Substituting for δ in equation (20), we now have a first order d.e. for $\dot{\eta}$, i.e.

$$\dot{\eta} = \frac{K_{A4}}{\tau}(\delta - K_{A5}\, a_1 \alpha - K_{A5}\, a_2 \eta - K_{A2}\, z - K_{A3}\, v) - \frac{\eta}{\tau}$$

Thus all the equations of the height control system are now defined as a set of first order differential equations.

The partitioning scheme is such that there are no integrations performed digitally. At time, t_{N-1}, the values for α and η are passed to the digital computer for evaluation of $(\dot{v})_N$ and $(\dot{q})_N$ in the interval $t_{N-1} < t < t_N$. These derivatives will be integrated in the interval $t_N < t < t_{N+1}$ on the analogue computer. We will assume, for the purposes of defining $A_1, \ldots, A_4$, $D_1, \ldots, D_4$ that the coefficients, $\frac{a_1}{u}$, $\frac{K_{A4}\, K_{A5}\, a_1}{\tau}$, $\frac{a_2}{u}$, $\frac{K_{A4}\, K_{A5}\, a_2}{\tau}$ are computed digitally in the interval $t_{N-1} < t < t_N$.

The equations defining the partitioning scheme are

$$\dot{\alpha} = \frac{a_1}{u}(\alpha)_{N-1} + q + \frac{a_2}{u}(\eta)_{N-1}$$

$$\dot{q} = a_3 (\alpha)_{N-1} + a_4 (\eta)_{N-1}$$

$$\dot{\beta} = \frac{K_{A1}}{T_{A2}} q - \frac{1}{T_{A2}} \beta$$

$$\dot{\eta} = \frac{-K_{A4}\, K_{A5}\, a_1}{\tau} (\alpha)_{N-1} - \frac{1}{\tau}\eta - \frac{K_{A4} K_{A5} a_2}{\tau} (\eta)_{N-1} - \frac{K_{A4}\, K_{A2}}{\tau} z - \frac{K_{A4}\, K_{A3}}{\tau} v + \frac{K_{A4}}{\tau}\delta$$

$$\dot{z} = v$$

$$\dot{v} = a_1 (\alpha)_{N-1} + a_2 (\eta)_{N-1}$$

$$\dot{\delta} = \frac{T_{A1}}{T_{A3}} \frac{K_{A1}}{T_{A2}} q + \frac{1}{T_{A3}}\left(1 - \frac{T_{A1}}{T_{A2}}\right)\beta - \frac{1}{T_{A3}}\delta$$

We have

$$\mathcal{X} = \begin{pmatrix} \alpha \\ q \\ \beta \\ \eta \\ z \\ v \\ \delta \end{pmatrix} \text{ and } \mathcal{Z} = \mathcal{Q}$$

In the interval $t_N < t < t_{N+1}$, we have

$$\dot{\chi} = \begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\beta} \\ \dot{\eta} \\ \dot{z} \\ \dot{v} \\ \dot{\delta} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{K_{A1}}{T_{A2}} & \frac{-1}{T_{A2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau} & \frac{-K_{A4}\,K_{A2}}{\tau} & \frac{-K_{A3}\,K_{A4}}{\tau} & \frac{K_{A4}}{\tau} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{T_{A1}}{T_{A3}}\frac{K_{A1}}{T_{A2}} & \frac{1}{T_{A3}}(1-\frac{T_{A1}}{T_{A2}}) & 0 & 0 & 0 & \frac{-1}{T_{A3}} \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \beta \\ \eta \\ z \\ v \\ \delta \end{pmatrix}$$

$$+ \begin{pmatrix} \frac{a_1}{u} & 0 & 0 & \frac{a_2}{u} & 0 & 0 & 0 \\ a_3 & 0 & 0 & a_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-K_{A4}\,K_{A5}\,a_1}{\tau} & 0 & 0 & \frac{-K_{A4}\,K_{A5}\,a_2}{\tau} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_1 & 0 & 0 & a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \beta \\ \eta \\ z \\ v \\ \delta \end{pmatrix}_{N-1}$$

i.e.

$$\dot{\chi} = A_1 \chi + D_1 \chi_{N-1}$$

The matrices A_2, D_2, A_3, D_3, A_4 and D_4 are all null.
Consider the situation where

$$a_1 = -193.8$$

$$a_2 = -34.5$$

$$a_3 = -58.4$$

$$a_4 = -61.6$$

$$u = 306$$

$$K_{A1} = 39$$

$$K_{A2} = 2.17$$

$$K_{A3} = 2.17$$

$$K_{A4} = 0.07$$

$$K_{A5} = 0.126$$

$$\tau = 0.001$$

$$T_{A1} = 0.5$$

$$T_{A2} = 2.0$$

$$T_{A3} = 0.01$$

Thus the data for defining the configuration for the software package is

$$K = 7$$

$$M = 8$$

$$M-K = 1$$

A_1, D_1 are 7 x 7 matrices

$A_2, \ldots, D_4$ are all null, and

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 19.5 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1000 & -151.9 & -151.9 & 70 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 975.0 & 75.0 & 0 & 0 & 0 & -100.0 \end{pmatrix}$$

and

$$D_1 = \begin{pmatrix} -0.6 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ -58.4 & 0 & 0 & -61.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1709.3 & 0 & 0 & +304.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -193.8 & 0 & 0 & -34.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Table 3 lists the input data for the software package and Appendix VII presents computer output for the above example. As is to be expected the partitioning scheme is shown to be far from ideal, and the results show that it would be mathematically unstable for a step-length as small as 12 ms, even though the bandwidth of the acceleration and height loops are 0.9 Hz and 0.2 Hz respectively. The general experience is that even the use of prediction will not greatly help in these sorts of application which employ "complete partitioning".

TABLE 3. INPUT DATA FOR EXAMPLE 2

Parameter value	Data Record
K,M	7,8
NEL	11
A1	1,2,1.0,3,2,19.5,7,2,975.0,3,3,-0.5, 7,3,75.0,4,4,-1000.0,4,5,-151.9,4,6,-151.9, 5,6,1.0,4,7,70.0,7,7,-100.0
NEL	8
D1	1,1,-0.6,2,1,-58.4,4,1,1709.3,6,1,-193.8, 1,4,-0.1,2,4,-61.6,4,4,304.3,6,4,-34.5
NEL (A2)	0
NEL (D2)	0
NEL (A3)	0
NEL (D3)	0
NEL (A4)	0
NEL (D4)	0
NT, EL, HINT	5,0.009,0.001
IMETH	2

A better partitioning scheme is one for which a_z and $\dot{q}$ are evaluated on the analogue computer, and the inertial platform is simulated digitally. The software package was used to determine the stability of this partitioning scheme for the numerical values of the previous example (but with $a_2 = 0$). It was predicted by the software package that instability would occur for a digital computing step-length of 113 ms, which is a considerable improvement over the 12 ms of the first scheme. This value was checked against the hybrid simulation by incorporating a variable delay in the digital computer. For the same values it was determined experimentally that instability occurred for a step-length of 110 ms, which is in close agreement with that predicted by the software package. In actual fact the hybrid simulation was inaccurate at this step-length, even though stable, highlighting the need for future work to consider the accuracy of hybrid simulation.

Finally, by using a simplified description of the above partitioning scheme, the question of stability can be settled by direct analysis, as in Section 5.1 of reference 1. For values of K_1, $K_2 = 6.6$ and $T = 0.2$, and following the method of reference 1, the maximum value of h for which the partitioning scheme is stable is 120 ms. Thus there is a good measure of agreement between all approaches.

6. TEST STATUS OF THE SOFTWARE PACKAGE

The package was extensively tested during each stage of its production. However, testing the complete package posed a problem since a typical system for analysis is likely to be quite complex, and one can expect the D matrix to be of the order 14 x 14. Finding the eigenvalues of this matrix by hand is quite intractable.

However, it is possible, by reducing the time constants in the response functions of the analogue portion, to generate 'instantaneous' responses for that part, and thus produce a system which can be analysed theoretically. One can therefore theoretically analyse a partitioning scheme for such a system, and hence, produce a value for step-length at which it will become unstable.

Such modified system equations can be used as the test data for the software package.

The homing missile simulation of the first example was chosen as a test case. The configuration is changed slightly with the dynamics of the radar servo and pedestal system now being represented on the analogue computer. The configuration is shown in figure 7.

The data defined in Section 5.1 is applied with the time constants of the transfer functions being increased by an order of 10. The equations for the instantaneous system are

Digital

$$\dot{\psi}_I = -3(\psi_{DM} + \psi_M)$$

$$= \gamma$$

Analogue

$$\psi_{DM} = \frac{(1 + 0.01\ S)(\psi_I - \psi_M)}{1 + 0.022\ S + 0.00016\ S^2}$$

$$r = \frac{3.52(1 + 0.001\ S)(1 + 0.19\ S)\ \gamma}{(1 + 0.0023\ S)(1 + 0.054\ S + 0.00094\ S^2)}$$

$$\dot{\psi}_M = r$$

These equations reduce to seven first order d.e.'s with the analogue vector, $\underset{\sim}{y}$, being defined by

$$\underset{\sim}{y} = \begin{pmatrix} \psi_{DM} \\ \psi_M \\ r \\ z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

and the digital vector, $\underset{\sim}{z}$, being defined by a single element.

$$\underset{\sim}{z} = (\psi_I)$$

The partitioned system can now be defined in terms of the matrices suitable for the software package.

In vector form the system is defined for the interval $t_N < t < t_{N+1}$

$$\dot{\chi} = \begin{pmatrix} \dot{\psi}_{DM} \\ \dot{\psi}_{M} \\ \dot{r} \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{pmatrix} = \begin{pmatrix} -137.5 & -62.5 & 0 & -6250 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -57.4 & 0 & 114.3 & -1064 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -43.5 & 0 \\ 0 & 0 & 1 & 0 & -0.6 & 0 \end{pmatrix} \begin{pmatrix} \psi_{DM} \\ \psi_{M} \\ r \\ z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

$$+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -927.5 & -927.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -4590 & -4590 & 0 & 0 & 0 & 0 \\ 4.6 & 4.6 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi_{DM} \\ \psi_{M} \\ r \\ z_1 \\ z_2 \\ z_3 \end{pmatrix}_{N-1}$$

$$+ \begin{pmatrix} 62.5 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} (\psi_I)_N$$

and

$$\dot{z} = (\dot{\psi}_I) = (-3 \ -3 \ 0 \ 0 \ 0 \ 0) \begin{pmatrix} \psi_{DM} \\ \psi_{M} \\ r \\ z_1 \\ z_2 \\ z_3 \end{pmatrix}_N$$

In terms of the matrices $A_1, \ldots, A_4, D_1, \ldots, D_4$,

$$\dot{\chi} = A_1 \chi + D_1 \chi_{N-1} + A_2 z_N$$

with D_2 null, and

$$\dot{z} = D_3 \chi_N$$

with A_3, A_4, D_4 null.

Solving the stability problem theoretically, we have on the digital computer in the interval $t_N < t < t_{N+1}$

$$\psi_I^{N+1} = \psi_I^N + 3h(-\psi_{DM}^N - \psi_M^N)$$

using Euler integration.

In the same interval the analogue functions are computed so fast as to be considered instantaneous. We can therefore represent the analogue functions by the following set of equations.

$$r = -3.52 \times 3(\psi_{DM}^{N-1} + \psi_M^{N-1})$$

$$\dot{\psi}_M = r$$

$$\psi_{DM} = \psi_I^N - \psi_M$$

At time, t_{N+1}

$$\psi_I^{N+1} = \psi_I^N - 3h(\psi_{DM}^N + \psi_M^N) \tag{21}$$

$$\psi_M^{N+1} = \psi_M^N - 3.52 \times 3h(\psi_{DM}^{N-1} + \psi_M^{N-1}) \tag{22}$$

$$\psi_{DM}^{N+1} = \psi_I^N - \psi_M^{N+1} \tag{23}$$

from equation (23), we have

$$\psi_I^N = \psi_{DM}^{N+1} + \psi_M^{N+1} \tag{24}$$

Thus

$$\psi_I^{N+1} = \psi_I^N - 3h\ \psi_I^{N-1} \tag{25}$$

and

$$\psi_M^{N+1} = \psi_M^N - 3h(\psi_I^{N-2}) \tag{26}$$

Equations (25) and (26) are recurrence relations, which together define the complete system. It is clear that the stability of equation (26) depends on the stability of equation (25), i.e. stability is determined from

$$\psi_I^{N+1} = \psi_I^N - 3h\ \psi_I^{N-1}$$

Clearly a solution to equation (25) is

$$\psi_I^N = a_1 \beta_1^N + a_2 \beta_2^N \tag{27}$$

where β_1 and β_2 are the roots of

$$\beta^2 - \beta + 3h = 0 \tag{28}$$

i.e.

$$\beta_{1,2} = \frac{1 \pm \sqrt{1 - 12h}}{2}$$

If $h < \frac{1}{12}$, the roots are real, with

$$|\beta_{1,2}| < 1$$

Hence equation (27) is a stable solution to equation (25).

If $h > \frac{1}{12}$, the roots are complex conjugates, and for stability of equation (27), their product must be in modulus less than unity, i.e. from equation (28) we need

$$3h < 1$$

which means, for stability we require

$$h < \frac{1}{3} s$$

The software package was run, and predicted the system would go unstable at exactly

$$h = 0.33 s$$

7. CONCLUSIONS

The analysis of reference 1 has shown that the problem of stability of a given hybrid simulation can be solved in advance of programming, provided one can find a linear approximation to the response of the system under consideration. For large-sized problems, however, the steps involved in applying the direct analysis may be intractable.

This report describes a software package which implements the stability analysis procedure, thus providing hybrid computer users a quick and flexible means of determining in advance the stability of a proposed partitioning scheme. In addition, the report attempts to provide guidelines for interpreting results obtained by the software package.

8. ACKNOWLEDGEMENT

The author would like to thank Mr M.J. Fitzpatrick of Electronic Warfare Studies Group for his assistance in the programming task.

NOTATION

A	MXM matrix describing linear system response
A_1, A_2, A_3, A_4	matrices used in partitioning of A
B	matrix used in partitioning of A, also inverse matrix of eigenvectors of A_1
C	matrix used in partitioning of A_1 also matrix of eigenvectors of A_1
D	stability matrix
D_1, D_2, D_3, D_4	matrices used in partitioning of A
$F(S)$ $G(S)$	transfer functions used in example
G	response matrix for system programmed on analogue portion
$H(S)$	transfer function used in example
I_K	unit matrix of order K
K, K_{A1}, K_{A2} K_{A3}, K_{A4}, K_{A5}	gains used in examples
N_o	number of integration steps in interval (o,h)
$P(S)$	transfer function used in example
Q	response of the analogue system to unit step input
$T_{A1}, T_{A2}, T_{A3},$ $T_{A4}, T_{A5},$ T_{D1}, T_{D2}, T_{D3}	time constants used in examples
$X(t)$	unique fundamental matrix
z	missile height
z_D	demanded missile height
z_1, z_2, z_3	used in example to reduce n^{th} order d.e. to n first order d.e.'s
a	maximum element in matrix A^3
a_D	demanded missile acceleration
a_Z	missile acceleration
a_1, a_2, a_3, a_4	functions of several variables related to missile incidence, Mach number and airpressure
h, h_s, h_o	hybrid simulation step-length

h_S, h_{STEP}	numerical integration step-length
q	missile pitch rate
r	missile yaw rate
t, t_N	time
u	missile velocity
$\mu(t)$	system input
v	missile vertical velocity
$\left.\begin{matrix}\mathfrak{X}\\ \mathfrak{Y}\\ \mathfrak{Z}\end{matrix}\right\}$	vectors used in describing system state
Γ_N	error function for fundamental matrix
Θ_K	null matrix of order K
$\Phi(t)$	fundamental matrix
ψ	equivalent to e^{At}
α	pitch incidence in example
β	used in example to reduce n^{th} order to d.e. to n first order d.e.'s
β_1, β_2	zeros of stability quadratic in example
γ	receiver output in example
δ	used in example to reduce n^{th} order d.e. to n first order d.e.'s
ϵ	accuracy of numerical integration solution
η	pitch wing angle in example
$\overline{\eta}$	pitch autopilot error signal in example
λ_i	eigenvalue of partitioning matrix A_1
μ_i	eigenvalue of stability matrix D
τ	time increments for numerical integration interval (o,h)
ψ_{DM}	radar dish angle in example
ψ_I	integral of receiver output in example
ψ_M	missile attitude

REFERENCES

No.	Author	Title
1	Allison, J.S. and Johnson, H.M.	"On the Stability of Hybrid Simulation". WRE-TN-1722 (AP), November 1976
2	Bekey, G.A. and Karplus, W.	"Hybrid Computation". Wiley, 1968
3	Johnson, H.M.	"A Mathematical Model of a Sea Skimming Missile". WRE-TN-1717 (AP), November 1976
4	Ogata, K.	"State Space Analysis of Control Systems". Prentice-Hall Inc., 1967

APPENDIX I

ANALYSIS OF THE GENERAL PARTITIONING PROBLEM

Consider the problem of solving, by hybrid computation, the system of differential equations

$$\dot{\underline{x}} = B\underline{x} + C\underline{u}(t) \tag{I.1}$$

where $\underline{x}$ is a vector of order M, and $\underline{u}(t)$ is a vector function of time. This set may represent the whole, or only part of the system being simulated, but it is sufficient in the present context for investigating the stability of integration processes.

For the purposes of stability analysis, one can disregard the forcing function, and set

$$\underline{u}(t) = \underline{0}$$

in equation (I.1), and henceforth be concerned with the system

$$\dot{\underline{x}} = B\underline{x} \tag{I.2}$$

The general solution to solving equation (I.2) on a hybrid computer is to allow for integrations and evaluation of derivatives to be performed digitally or on the analogue computer without restriction. The range of possibilities, then, is as shown in figure 1.

If K integrations are carried out on the analogue computer and M-K on the digital portion, then equation (I.2) can be re-written in the form

$$\begin{aligned} \dot{\underline{y}} &= A_1\underline{y} + A_2\underline{z} + D_1\underline{y} + D_2\underline{z} \\ \dot{\underline{z}} &= A_3\underline{y} + A_4\underline{z} + D_3\underline{y} + D_4\underline{z} \end{aligned} \tag{I.3}$$

where

$$\underline{x} = \begin{pmatrix} \underline{y} \\ \underline{z} \end{pmatrix}$$

and the vectors $A_1\underline{y}$, $A_2\underline{z}$, $A_3\underline{y}$, $A_4\underline{z}$ are computed on the analogue portion while $D_1\underline{y}$, $D_2\underline{z}$, $D_3\underline{y}$ and $D_4\underline{z}$ are determined digitally. The matrix B as defined in equation (I.2) is then assumed partitioned such that

$$B = A + D,$$

where A, D are matrices comprising those elements of B which are evaluated on the analogue and digital computers respectively. Further A and D are assumed partitioned such that

$$A = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}$$

and

$$D = \begin{pmatrix} D_1 & D_2 \\ D_3 & D_4 \end{pmatrix}$$

For the purpose of this analysis, the digital to analogue, and analogue to digital interfaces are assumed to be sampled simultaneously.

Assume $\underline{x}$ is sampled by the analogue to digital converter at time, t_N, and the digital computer step-length for each cycle is h seconds. Then at time

$$t_{N+1} = t_N + h$$

the quantity z_{N+1} is fed through the digital to analogue interface for determining $\underline{x}$ in the interval $t_{N+1} < t < t_{N+2}$.

Euler integration on the digital computer gives from (I.3)

$$z_{N+1} = z_N + h(A_3 x_N + A_4 z_{N-1} + D_3 x_N + D_4 z_N) \qquad (I.4)$$

In the same interval, $t_N < t < t_{N+1}$, $\dot{\underline{x}}$ can be integrated on the analogue by solving

$$\dot{x} = A_1 x + A_2 z_N + D_1 x_{N-1} + D_2 z_N \qquad (I.5)$$

In equation (I.5) $A_1 x$ varies with time, while

$$A_2 z_N + D_1 x_{N-1} + D_2 z_N$$

is constant over the interval $t_N < t < t_{N+1}$.

The solution to equation (I.5) is given by (see Appendix V)

$$x_{N+1} = \Phi(h)\ \Phi^{-1}(o)\ x_N + \int_o^h \Phi(\xi)\ \Phi^{-1}(o)\ d\xi\ (A_2 z_N + D_1 x_{N-1} + D_2 z_N) \qquad (I.6)$$

which reduces to

$$x_{N+1} = G\ x_N + Q(A_2 z_N + D_1 x_{N-1} + D_2 z_N) \qquad (I.7)$$

where

$$G = \Phi(h)\ \Phi^{-1}(o)$$

$$Q = \int_o^h G(\xi)\ d\xi$$

and $\Phi(t)$ is a fundamental matrix for A_1.

Equations (I.4) and (I.7) can be combined into

$$\begin{pmatrix} Y_{N+1} \\ Z_{N+1} \end{pmatrix} = \begin{pmatrix} G & Q(A_2 + D_2) \\ h(A_3 + D_3) & I_{M-k} + hD_4 \end{pmatrix} \begin{pmatrix} Y_N \\ Z_N \end{pmatrix} + \begin{pmatrix} QD_1 & \Theta_1 \\ \Theta_2 & hA_4 \end{pmatrix} \begin{pmatrix} Y_{N-1} \\ Z_{N-1} \end{pmatrix} \qquad (I.8)$$

where

Θ_1 is a null matrix of order Kx(M-K)

Θ_2 is a null matrix of order (M-K)xK

Setting

$$Y_N = \begin{pmatrix} QD_1 & \Theta_1 \\ \Theta_2 & hA_4 \end{pmatrix} \begin{pmatrix} Y_{N-1} \\ Z_{N-1} \end{pmatrix}$$

equation (I.8) becomes

$$X_{N+1} = \begin{pmatrix} G & Q(A_2 + D_2) \\ h(A_3 + D_3) & I_{M-K} + hD_4 \end{pmatrix} X_N + Y_N$$

$$Y_{N+1} = \begin{pmatrix} QD_1 & \Theta_1 \\ \Theta_2 & hA_4 \end{pmatrix} X_N \qquad (I.9)$$

This system can be further transformed by setting

$$u_N = \begin{pmatrix} X_N \\ Y_N \end{pmatrix}$$

and then equation (I.9) reduces to

$$u_{N+1} = \left(\begin{array}{cc|c} G & Q(A_2 + D_2) & \\ h(A_3 + D_3) & I_{M-k} + hD_4 & I_M \\ \hline QD_1 & \Theta_1 & \\ \Theta_2 & hA_4 & \Theta_M \end{array}\right) u_N \qquad (I.10)$$

$$= D\, u_N \qquad (I.11)$$

Thus, the chosen computing scheme will be stable if the zeros of the $2M^{th}$ order polynomial

$$f(\lambda) = |D - \lambda I_{2M}| \qquad (I.12)$$

are in modulus less than unity.

APPENDIX II

NUMERICAL INTEGRATION STEP-LENGTH FOR DETERMINING $\Phi(h)$

It is required to determine the solution to the fundamental matrix equation using numerical integration over the interval $(0,h_o)$, i.e.,

$$\dot{\Phi}(h_o) = B\Phi(h_o)$$

where

$$\Phi(o) = I$$

and

B is a real MXM matrix.

The accuracy and stability of the numerical solution depend on the numerical integration step-length, h, which will be determined from the eigenvalues of B. The numerical integration method chosen is the predictor-corrector method.

For the predictor-corrector method at the $(N+1)^{th}$ step we have,

Predict pass:

$$\Phi^P_{N+1} = \Phi^P_N + hB\Phi_N$$

If the eigenvalues of B are distinct, then we can find a matrix P, such that

$$P\,B^{-1}\,P = \text{diag}\,(\lambda_1,\ldots,\lambda_M)$$

where λ_i are the eigenvalues of B.

(If the λ_i are not all distinct we need only consider the reduced set of recurrence relations for which the eigenvalues are distinct).

Setting

$$\Psi_N = P\Phi_N$$

we obtain

$$\Psi^P_{N+1} = \Psi^P_N + h\,\text{diag}\,(\lambda_1,\ldots,\lambda_M)\,\Psi_N$$

For the k^{th} column we have for the predict pass

$$\Psi^P_{N+1} = \Psi^P_N + h\,\lambda_k\,\Psi_N$$

$$\dot{\Psi}_{N+1} = \lambda_K\,\Psi^P_{N+1}$$

Correct pass:

$$\Psi_{N+1} = \Psi_N + \frac{h}{2}(\dot{\Psi}_N + \dot{\Psi}_{N+1})$$

$$= \Psi_N + h\lambda_k \Psi_N + \frac{h^2 \lambda^2_k}{2} \Psi_N$$

The question of stability can be determined by analysis of the stability of the set of recurrence relations

$$\Psi^k_{N+1} = \left(1 + \lambda_k h + \frac{\lambda_k^2 h^2}{2}\right) \Psi^k_N, \; k = 1,\ldots,M \qquad \text{(II.1)}$$

Putting

$$\beta^N = \Psi_N$$

we have

$$\beta = 1 + \lambda_k h + \frac{\lambda_k^2 h^2}{2}$$

and require

$$|\beta| < 1$$

for a stable solution.

Hence the stability boundary is defined by

$$|f(h)| = \left|1 + \lambda_k h + \frac{\lambda_k^2 h^2}{2}\right| = 1, \; k = 1,\ldots, M \qquad \text{(II.2)}$$

and is shown in figure 8.

It is necessary for the accuracy and stability required in the present application for h to be chosen so that $\lambda_k h$ lies well inside the stability boundary for all k.

Consider the real part of the eigenvalues, i.e.,

$$u = Re(\lambda_k)$$

Then we have stability if,

$$uh < -2$$

In order to achieve uh well inside the stability boundary we choose h so that we have

$$uh = -0.2 \qquad \text{(II.3)}$$

To further ensure accuracy and stability it is necessary that the above value of h satisfies

$$\frac{vh}{w_B(h)} < 0.2 \qquad \text{(II.4)}$$

where

$$v = \text{Im}(\lambda_k)$$

$$w_B(h) = \sqrt[4]{-8uh} \text{ (see below)}$$

If equation (II.4) is not satisfied, then it is necessary to define a new value of numerical integration step-length, h_1, such that h_1 satisfies the accuracy and stability criteria. We require an h_1 such that uh_1 are small, and vh_1 lie well inside the stability boundary for all the eigenvalues of B.

Putting

$$\lambda_k = u + iv$$

in equation (II.2), we have for the stability contour

$$f(h) = 1 + uh + \frac{(u^2 - v^2)}{2} h^2 + i\, vh(1 + uh)$$

$$|f|^2 = (1 + uh)^2 + 2(1 + uh)(u^2 - v^2)\frac{h^2}{2} + \frac{(u^2 - v^2)^2}{4} h^4 + v^2h^2(1 + uh)^2$$

$$= 1 + 2uh + 2u^2h^2 + u^3h^3 + \frac{u^4h^4}{4} + v^2uh^3 + \frac{v^2u^2h^4}{2} + \frac{v^4h^4}{4}$$

Since

$$|f| = 1,$$

we have

$$uh\left(2 + 2uh + u^2h^2 + \frac{u^3h^3}{3} + v^2h^2 + \frac{v^2uh^3}{4}\right) + \frac{v^4h^4}{4} = 0 \qquad \text{(II.5)}$$

For complex eigenvalues we are interested in those for which uh is small. Thus equation (II.5) reduces to

$$2uh + v^2h^2\, uh + \frac{v^4h^4}{4} = 0$$

Setting

$$a = uh$$

we have

$$2a + v^2h^2a + \frac{v^4h^4}{4} = 0$$

$$(vh)^2 = \frac{-4a \pm \sqrt{16a^2 - 32a}}{2}$$

$$= 2|a| \pm \sqrt{8|a| + 4a^2}$$

$$vh = \sqrt{2|a| \pm \sqrt{8|a| + 4a^2}}$$

Hence for uh small, vh is also small, and in equation (II.5) we can ignore the term in v^2h^2, but not the 4^{th} order term since it is independent of uh. Thus the stability boundary in equation (II.5) can be expressed by the relationship

$$2uh \approx \frac{-v^4h^4}{4}$$

Suppose we choose a step-length h_1, such that

$$w = vh_1$$

On the boundary we have

$$w_B = \sqrt[4]{-8uh_1}$$

We require the step-length to be chosen so that w is well inside the boundary, thus we choose h_1 such that

$$\frac{w}{w_B(h_1)} < 0.2$$

i.e., we have

$$\frac{vh_1}{\sqrt[4]{-8uh_1}} = 0.2$$

Hence, h_1 is chosen by application of the expression

$$h_1 = \sqrt[3]{\frac{-u}{v^4}}\,\sqrt[3]{8 \times (0.2)^4}$$

APPENDIX III

LISTING OF SOFTWARE PACKAGE

```
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON O(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
       DIMENSION IIK(10),IK(10),DUM(100),X(10,10)
C  READ DIMENSION OF MATRICES DEFINING THE PARTITIONING SCHEME
      READ(5,*)K,M
      MMK=M-K
      MTWO=M+M
C  ZERO MATRICES A1,....,D4.
      DO 3 I=1,K
      DO 2 J=1,K
      A1(I,J)=0.
    2 D1(I,J)=0.
      DO 3 J=1,MMK
      A2(I,J)=0.
      D2(I,J)=0.
      A3(J,I)=0.
    3 D3(J,I)=0.
      DO 4 I=1,MMK
      DO 4 J=1,MMK
      A4(I,J)=0.
    4 D4(I,J)=0.
C  READ NON-ZERO ELEMENTS OF A1,.....,D4.
      READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 5
      READ(5,*)(IIK(L),IK(L),A1(IIK(L),IK(L)),L=1,NEL)
    5 READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 6
      READ(5,*)(IIK(L),IK(L),D1(IIK(L),IK(L)),L=1,NEL)
    6 READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 7
      READ(5,*)(IIK(L),IK(L),A2(IIK(L),IK(L)),L=1,NEL)
    7 READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 8
      READ(5,*)(IIK(L),IK(L),D2(IIK(L),IK(L)),L=1,NEL)
    8 READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 9
      READ(5,*)(IIK(L),IK(L),A3(IIK(L),IK(L)),L=1,NEL)
    9 READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 11
      READ(5,*)(IIK(L),IK(L),D3(IIK(L),IK(L)),L=1,NEL)
   11 READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 12
      READ(5,*)(IIK(L),IK(L),A4(IIK(L),IK(L)),L=1,NEL)
   12 READ(5,*) NEL
      IF(NEL.EQ.0)GO TO 13
      READ(5,*)(IIK(L),IK(L),D4(IIK(L),IK(L)),L=1,NEL)
   13 READ(5,*)  NT,EL,HINT
      READ(5,*)IMETH
```

```
C  COMPUTE OR READ THE RANGE OF H FOR TESTING STABILITY.
      DO 10 I=1,NT
   10 H(I)=EL + DFLOAT(I-1)*HINT
      IF(H(NT).EQ.0)READ(5,*)(H(I),I=1,NT)
C SET UP STEP LENGTHS
      IF(NT.NE.1)GO TO 18
      H(2)=1.5D0*H(1)
      NT=2
C  WRITE OUT INPUT DATA
   18 WRITE(6,19)
   19 FORMAT(2X,'INPUT DATA',/,2X,10(1H.),/)
      WRITE(6,15)K,M
      WRITE(6,21)K,K
   21 FORMAT(2X,'A1 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (A1,K,K,X)
      WRITE(6,22)K,K
   22 FORMAT(2X,'D1 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (D1,K,K,X)
      WRITE(6,23)K,MMK
   23 FORMAT(2X,'A2 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (A2,K,MMK,X)
      WRITE(6,24)K,MMK
   24 FORMAT(2X,'D2 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (D2,K,MMK,X)
      WRITE(6,25)MMK,K
   25 FORMAT(2X,'A3 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (A3,MMK,K,X)
      WRITE(6,26)MMK,K
   26 FORMAT(2X,'D3 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (D3,MMK,K,X)
      WRITE(6,27)MMK,MMK
   27 FORMAT(2X,'A4 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (A4,MMK,MMK,X)
      WRITE(6,28)MMK,MMK
   28 FORMAT(2X,'D4 MATRIX-(',I2,' X',I2,')')
      CALL OUTPUT (D4,MMK,MMK,X)
      WRITE(6,20)NT,(H(I),I=1,NT)
   15  FORMAT(2X,'K=',I2,2X,'M=',I2)
   20 FORMAT (2X,'THE ',I2,' STEPLENGTHS ARE-',/,10(2X,F6.4))
      WRITE(6,102)
      WRITE(6,14)IMETH
   14 FORMAT(2X,'IMETH=',I1)
       DO 16 I=1,K
       DO 16 J=1,K
   16  A11(I,J)=A1(I,J)
C  COMPUTE THE EIGENVALUES AND MATRIX OF EIGENVECTORS FOR A1
      WRITE(6,17)
   17 FORMAT(1H1)
      CALL EIGEN
      WRITE(6,102)
  102 FORMAT(1H )
       WRITE(6,105)
  105  FORMAT(10X,'EIGENVALUES OF A1 (COMPLEX ELEMENTS)',/,10X, 35(1H-))
       WRITE(6,110)(F(I),I=1,K)
  110  FORMAT(4(2X,'(',E11.4,',',E11.4,')'),/)
      WRITE(6,125)
      WRITE(6,102)
C  TEST THE ZEROS OF A1,IE WHETHER THE E/VALUES HAVE
C  REAL PART NON-POSITIVE
```

```
      IST=0
      IMST=0
      INST=0
      DO 36 I=1,K
      IF(JCOUNT(I)-1)31,33,35
   31 IST=IST+1
      GO TO 36
   33 IMST=IMST+1
      GO TO 36
   35 INST=INST+1
   36 CONTINUE
      IF(INST.NE.0)GO TO 60
      IF(IMST.NE.0)GO TO 50
      WRITE(6,40)
   40 FORMAT(2X, 64HZEROS OF A1 HAVE -VE REAL PART.G MATRIX WILL BE STAB
     1LE FOR ALL H  )
        WRITE(6,102)
      GO TO 70
   50 WRITE(6,55) IMST
   55 FORMAT(2X,I2,74H OF THE ZEROS OF A1 HAVE ZERO REAL PART.G MATRIX W
     1ILL BE MARGINALLY STABLE  )
        WRITE(6,102)
      GO TO 70
   60 WRITE(6,65) INST,IMST
   65 FORMAT(2X, 9HTHERE ARE,I3,29H ZEROS WITH +VE REAL PART,AND,I3,56HZ
     1EROS WITH ZERO REAL PART.THUS G MATRIX WILL BE UNSTABLE  )
        WRITE(6,102)
      IMETH=1
      IF(U.GE.0.D0)GO TO 70
      WRITE(6,82)
      STOP
   70 CONTINUE
C FOR DISTINCT EIGENVALUES THE FUNDAMENTAL MATRIX
C IS COMPUTED USING THE MATRIX OF EIGENVECTORS OF A1,
C OR USING NUMERICAL INTEGRATION-   USER CHOICE.
C FOR REPEATED EIGENVALUES THE FUNDAMENTAL MATRIX
C IS COMPUTED USING NUMERICAL INTEGRATION ONLY
        IF(IEIG.EQ.0)GO TO 71
        WRITE(6,75)
   75 FORMAT(2X,35HTHE EIGENVALUES OF A1 ARE REPEATED
     1'G AND Q WILL BE OBTAINED BY NUMERICAL INTEGRATION' )
        WRITE(6,102)
      IF(INST.LE.0)GO TO 72
      WRITE(6,82)
   82 FORMAT(2X,'BOTH N.I AND DIRECT METHOD FOR COMPUTING FUNDAMENTAL'
     1,' MATRIX WILL FAIL')
      STOP
   71 WRITE(6,73)
   73 FORMAT(2X,'THE EIGENVALUES OF A1 ARE DISTINCT')
      WRITE(6,102)
   72 LL=NT
      IF(IMETH.EQ.1)GO TO 74
      CALL LAMBDA
   74 CONTINUE
        IPASS=0
      DO 100 IN=1,LL
      TIME=H(IN)
   80 IF(IMETH.EQ.2)GO TO 84
C  COMPUTE THE FUNDAMENTAL MATRIX FOR A1,AND THE MATRICES
```

```
C  G AND Q FOR THE RANGE OF H SPECIFIED AS INPUT
      CALL FUND
      WRITE(6,102)
      WRITE(6,1114)TIME
 1114  FORMAT(3X,'H=',F6.4,/)
      GO TO 86
C COMPUTE G(H),Q(H) BY NUMERICAL INTEGRATION
  84  CALL NUMERI(IPASS,IN,HINT)
      WRITE(6,102)
       WRITE(6,93)TIME,HS
  93  FORMAT(2X,'H=',F6.4,' NUMERICAL INTEGRATION STEPLENGTH IS',F9.6/)
  86   WRITE(6,115)
  115  FORMAT(10X,  'G MATRIX')
      DO 132 II=1,K
       WRITE(6,111)II,(G(II,J),J=1,K)
 111   FORMAT(1H0,2X,'ROW ',I2,10(1X,E11.4))
 132  CONTINUE
       WRITE(6,125)
       WRITE(6,112)
  112  FORMAT(10X,  'Q MATRIX')
      DO 133 II=1,K
       WRITE(6,111)II,(Q(II,J),J=1,K)
 133   CONTINUE
  125  FORMAT(1H ,120(1H-))
       WRITE(6,125)
C  COMPUTE THE MATRIX D OF ORDER 2M
      CALL DMATRX
       WRITE(6,130)
  130  FORMAT(10X,8HD MATRIX )
      DO 131 II=1,MTWO
      WRITE(6,113)II,(D(II,J),J=1,MTWO)
 113  FORMAT(1H0,2X,'ROW ',I2,10(1X,E11.4),/,9X,10(1X,E11.4))
 131   CONTINUE
       WRITE(6,125)
  135  FORMAT(1H ,120(1H*))
C  TEST THAT THE EIGENVALUES OF D ARE IN MODULUS LESS THAN UNITY
      CALL MODEIG
       WRITE(6,140)
  140  FORMAT(10X,'EIGENVALUES OF D (COMPLEX ELEMENTS)',/,
     110X,35(1H-))
       WRITE(6,110)(F(II),II=1,MTWO)
      IF(KCOUNT.GT.0)GO TO 87
      WRITE(6,85)TIME
   85 FORMAT(2X,  'ALL THE EIGENVALUES OF D HAVE MODULUS LESS THAN OR EQ
     1UAL TO UNITY FOR H=',F6.4,' SCHEME IS STABLE FOR CHOICE OF H')
      WRITE(6,135)
      GO TO 100
   87 WRITE(6,90)TIME,KCOUNT
   90 FORMAT(2X, 6HFOR H=F6.4,1H,I3,  ' OF THE EIGENVALUES OF D HAVE MOD
     1ULUS GREATER THAN UNITY.THEREFORE CHOICE IS UNSTABLE FOR THIS H')
      WRITE(6,110)(F(INDEX(II)),II=1,KCOUNT)
      WRITE(6,135)
  100 CONTINUE
      STOP
      END
      SUBROUTINE EIGEN
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     100(10,10),QFI(10,10),F(20),WW(150)
```

```
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
      COMPLEX*16 Y(10,10)
      CALL CNVRT (1,Y)
      CALL DIST
       IF (IMETH.EQ.2)RETURN
      IF(IEIG.EQ.0)GO TO 10
      IMETH=2
      RETURN
   10 DO 17 I=1,K
      DO 17 J=1,K
      W(1)=DREAL(C(I,J))
      W(2)=DIMAG(C(I,J))
      IF(DABS(W(1)).LT.1.D-14)W(1)=0.D0
      IF(DABS(W(2)).LT.1.D-14)W(2)=0.D0
   17 C(I,J)=DCMPLX(W(1),W(2))
   20 CALL CNVRT (2,Y)
      RETURN
      END
      SUBROUTINE DIST
      IMPLICIT  REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),W(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
      ID=0
      IDI=200
      IDIST=0
      DO 20 I=1,K
      JCOUNT(I)=0
      EI(I)=DIMAG(E(I))
      ER(I)=DREAL(E(I))
C TEST STABILITY BY CONSIDERING ZEROS OF A1.
      IF(ER(I).LT.0.D0)GO TO 20
      JCOUNT(I)=1
      IF(ER(I).EQ.0.D0)GO TO 20
      JCOUNT(I)=2
   20 CONTINUE
C TEST FOR REPEATED EIGENVALUES OF A1.
      IEIG=0
      IF(K.EQ.1)GO TO 35
      KM1=K-1
      DO 30 J=1,KM1
```

```
      JP1=J+1
      DO 30 I=JP1,K
       IF(ER(J).EQ.ER(I))ID=I
       IF(EI(J).EQ.EI(I))IDI=I
       IF(ID.EQ.IDI)IEIG=IEIG+1
   30 CONTINUE
   35  CONTINUE
      RETURN
      END
      SUBROUTINE FUND
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
C CREATE A DIAGONAL MATRIX
       ZERO=0.
      DO 20 I=1,K
      FI(I,I)=0.DO
      IF(ER(I).LT.0.D0)GO TO 6
      IF(EI(I).LT.0.D0)GO TO 6
   5  FI(I,I)=CDEXP(E(I)*TIME)
      GO TO 7
   6  IF(CDABS(E(I)*TIME).LT.179.D0)GO TO 5
   7   IF(CDABS(E(I)).EQ.ZERO)GO TO 10
      QFI(I,I)=(FI(I,I)-1.D0)/E(I)
      GO TO 15
  10   QFI(I,I)=TIME
  15  DO 20 J=1,K
      IF(J.EQ.I)GO TO 20
      FI(I,J)=0.
      QFI(I,J)=0.
   20 CONTINUE
      CALL DCMPY(C,K,K,FI,K,K,PHI)
      CALL DCMPY(PHI,K,K,B,K,K,GG)
C COMPUTE MATRICES G AND Q
      CALL DCMPY(C,K,K,QFI,K,K,PHI)
      CALL DCMPY(PHI,K,K,B,K,K,QQ)
       IMG=0
       IMQ=0
       DO 50 J=1,K
       DO 50 I=1,K
      DG=DIMAG(GG(I,J))
      DQ=DIMAG(QQ(I,J))
       IF(DABS(DG).GT.1.D-3)  IMG=IMG+1
       IF(DABS(DQ).GT.1.D-3)  IMQ=IMQ+1
  50   CONTINUE
       IF(IMG.GT.0)WRITE(6,60)IMG
       IF(IMQ.GT.0)WRITE(6,70)IMQ
  60   FORMAT(3X,I2,' OF THE IMAGINARY VALUES OF G MATRIX HAS A VALUE
     1 GREATER THAN 1E-3')
```

```
 70   FORMAT(3X,I2,' OF THE IMAGINARY VALUES OF Q MATRIX HAS A VALUE
     1 GREATER THAN 1E-3')
      DO 80 J=1,K
      DO 80 I=1,K
      G(I,J)=DREAL(GG(I,J))
 80   Q(I,J)=DREAL(QQ(I,J))
      RETURN
      END
      SUBROUTINE DMATRX
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
       MP1=M+1
       MP1=M+1
       MMK=M-K
       K1=K+1
       CALL  DMMPY(Q,K,K,D1,K,K,QD1)
       DO 50 J=1,K
       DO 15 I=1,K
       D(I,J)=G(I,J)
 15    D(I+M,J)=QD1(I,J)
       DO 20 L=K1,M
      I=L-K
      D(L,J)=TIME*(A3(I,J)+D3(I,J))
      D(L+M,J)=0.
 20   CONTINUE
      DO 30 I=1,MMK
 30   QD1(J,I)=  A2(J,I)+D2(J,I)
  50 CONTINUE
      CALL  DMMPY(Q,K,K,QD1,K,MMK,QD2)
      DO 70 J=1,MMK
      DO 60 I=1,K
      L=J+K
      D(I,L)= QD2(I,J)
 60    D(I+M,L)=0.
      DO 70 I=K1,M
       L=J+K
      II=I-K
      D(I,L)=TIME*D4(II,J)
       D(I+M,L)=TIME*A4(II,J)
 70    CONTINUE
       DO 80 I=K1,M
       D(I,I)=D(I,I)+1.
 80    CONTINUE
       DO 100 I=1,MTWO
       DO 90 J=MP1,MTWO
 90    D(I,J)=0.
       IF(I.GT.M)GO TO 100
       D(I,I+M)=1.
```

```
100   CONTINUE
      RETURN
      END
      SUBROUTINE MODEIG
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
      COMPLEX*16 Y(10,10)
       M2=MTWO
      KCOUNT=0
      ONE=1.DO
       DO 10 I=1,M2
10     INDEX(I)=0
      CALL CNVRT (3,Y)
      DO 20 I=1,M2
      IF(CDABS(F(I)).LE.ONE)GO TO 20
      KCOUNT=KCOUNT+1
      INDEX(KCOUNT)=I
 20 CONTINUE
      RETURN
      END
      SUBROUTINE CNVRT (I,Y)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
       COMPLEX*16 Y(K,K)
      IF(I-2)10,20,30
10     CALL EIGRF(A1, K,10,2,E,C,10,W,IER)
       WRITE(6,15)IER
15     FORMAT(2X,'ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE '
     1,'VALUE ',I3,/)
      IF(W(1).GE.1.DO)GO TO 17
      WRITE(6,16)
16    FORMAT(2X,'THE EIGENVALUE ROUTINE PERFORMED WELL',/)
      GO TO 40
17    IF(W(1).GT.100.DO)GO TO 19
      WRITE(6,18)
18    FORMAT(2X,'THE EIGENVALUE ROUTINE PERFORMED SATISFACTORILY',/)
      GO TO 40
```

```
 19   WRITE(6,21)
 21     FORMAT(2X,'THE EIGENVALUE ROUTINE PERFORMED POORLY',/)
      GO TO 40
 20    DO 2 II=1,K
       DO 2 J=1,K
  2    Y(II,J)=C(II,J)
       U=1.
       CALL MA23BD(Y,K,K,WW,U )
      IF(U.GE.0.D0)GO TO 3
      IMETH=2
      WRITE(6,80)
 80   FORMAT(2X,'MATRIX INVERSION ROUTINE HAS PRODUCED UNRELIABLE RESULT
     1S',/,2X,'THEREFORE NUMERICAL INTEGRATION METHOD WILL BE USED.',/)
      GO TO 40
 3    DO 5 II=1,K
      DO 5 J=1,K
 5    B(II,J)=Y(II,J)
       GO TO 40
 30    CALL EIGRF (D,MTWO,20,0,F,ZZ,10,W,IER)
      WRITE(6,15)IER
 40    RETURN
       END
       SUBROUTINE DMMPY(DPM1,MMM,NNN1,DPM2,NNN2,LLL,DPRES)
C                MATRIX MULTIPLICATION
C          DPM1      -- INPUT IS AN MMM X NNN1 MATRIX
C                       BUT IS ORDERED BY THE COMPUTER AS A 10*10
C          DPM2      -- INPUT IS A NNN2 X LLL MATRIX
C                       BUT IS ORDERED BY THE COMPUTER AS A 10*10
C          DPRES     -- OUTPUT IS AN 10 X 10 SQUARE MATRIX
C  N. B. INPUT MATRICES OF ORDER 2
C
       IMPLICIT  REAL*8(A-H),REAL*8(O-Z)
      DIMENSION DPM1(10,10),DPM2(10,10),DPRES(10,10)
C
C  CHECK THAT DIMENSIONS OK
       IF(NNN1.NE.NNN2) GO TO 70
C
      DO 50  K=1,LLL
C       OUTSIDE LOOP
      DO 50  I=1,MMM
C       INSIDE LOOP
C
C     BUILD EACH ELEMENT OF THE PRODUCT
      CTEMP=0.
      DO 40  J=1,NNN1
C
C       COLUMN*ROW
      CTEMP=CTEMP + DPM1(I,J)*DPM2(J,K)
 40    CONTINUE
C
 50   DPRES(I,K)=CTEMP
C
      RETURN
C
C  INPUTS NOT EQUAL
 70   WRITE(6,201)NNN1,NNN2
 201   FORMAT(1X,' DMMPY DIMENSIONS FED IN NOT COMPATIBLE FOR ',
     1 /,' MULTIPLICATION  --  EXECUTION TERMINATED',/,2X,'NNN1=',
     2 I3,5X,'NNN2=',I3)
```

```
      STOP
      END
      SUBROUTINE DCMPY(COMPL1,MMM,NNN1,COMPL2,NNN2,LLL,CMPRES)
C      COMPLEX MATRIX MULTIPLICATION
C        COMPL1      -- INPUT IS AN MMM X NNN1 COMPLEX MATRIX
C                       BUT IS ORDERED BY THE COMPUTER AS A 10*10
C        COMPL2      -- INPUT IS A NNN2 X LLL COMPLEX MATRIX
C                       BUT IS ORDERED BY THE COMPUTER AS A 10*10
C        CMPRES      -- OUTPUT IS AN 10 X 10 SQUARE COMPLEX MATRIX
C  N. B. INPUT MATRICES OF ORDER 2
C
      COMPLEX*16 COMPL1(10,10),COMPL2(10,10),CMPRES(10,10),CTEMP,CMPLX
C
C  CHECK THAT DIMENSIONS OK
      IF(NNN1.NE.NNN2) GO TO 70
C
      DO 50  K=1,LLL
C       OUTSIDE LOOP
      DO 50  I=1,MMM
C       INSIDE LOOP
C
C     BUILD EACH ELEMENT OF THE PRODUCT
      CTEMP=CMPLX(0.,0.)
      DO 40  J=1,NNN1
C
C      COLUMN*ROW
      CTEMP=CTEMP + COMPL1(I,J)*COMPL2(J,K)
  40   CONTINUE
C
  50  CMPRES(I,K)=CTEMP
C
      RETURN
C
C  INPUTS NOT EQUAL
  70  WRITE(6,201)NNN1,NNN2
  201  FORMAT(1X,' DCMPY DIMENSIONS FED IN NOT COMPATIBLE FOR ',
     1 /,' MULTIPLICATION  --  EXECUTION TERMINATED',/,2X,'NNN1=',
     2 I3,5X,'NNN2=',I3)
       STOP
       END
       SUBROUTINE NUMERI(IPASS,IN,HINT)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
C PERFORM NUMERICAL INTEGRATION PROCESSES TO
C DETERMINE MATRICES G AND Q.
      DIMENSION UNIT(10,10)
      IF(IPASS.NE.0)GO TO 40
C  INITIALIZE VARIABLES FOR SINT.
```

```
C  CALCULATE STEP-LENGTH FOR NUMERICAL INTEGRATION INTERVAL.
      LSTEP=0
      TAU=0.D0
      DO 10 I=1,K
      DO 5  J=1,K
    5 UNIT(I,J)=0.D0
   10 UNIT(I,I)=1.D0
      DO 15 I=1,K
      DO 15 J=1,K
      PFI(I,J)=UNIT(I,J)
      XINTA1(I,J)=UNIT(I,J)
      XINTA2(I,J)=0.D0
      DFI(I,J)=A11(I,J)
      YFI(I,J)=0.D0
      DYFI(I,J)=UNIT(I,J)
      YINTA1(I,J)=0.D0
   15 YINTA2(I,J)=0.D0
   40  IF(LSTEP.GT.0)GO TO 45
      CALL STEP (IN,HINT,LSTEP)
   45 NAM=-1
      IPASS=1
      TAU=TAU+HS
   50 IPASS=-IPASS
      CALL SINT(IPASS,NAM)
      IF(NAM)52,52,54
   52 CALL DMMPY(A11,K,K,PFI,K,K,DFI)
      GO TO 60
   54 DO 56 I=1,K
      DO 56 J=1,K
   56 DYFI(I,J)=PFI(I,J)
   60 IF(IPASS.LT.0)GO TO 50
      NAM=-NAM
      IF(NAM.GT.0)GO TO 50
      TAU=TAU+HS
      IF(TAU.LE.TIME)GO TO 50
      TAU=TAU-HS
      DO 70 I=1,K
      DO 70 J=1,K
      G(I,J)=PFI(I,J)
   70 Q(I,J)=YFI(I,J)
      RETURN
      END
       SUBROUTINE STEP(I,HINT,LSTEP)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
C ESTABLISH STEP LENGTH FOR NUMERICAL INTEGRATION FOR
C STABILITY AND ACCURACY OF NUMERICAL SOLUTION.
C SUPPLY STEPLENGTH,HS,FOR EACH INTERVAL (H(I+1),H(I)).
```

```
      IF(I.GT.2)GO TO 50
      IF(I.GT.1)GO TO 20
      HH=H(1)
      GO TO 21
   20 HH=H(I)-H(I-1)
   21 XVAL= HH /HSTEP
      NSTEP   =DINT(XVAL)+1
      HS    = HH /DFLOAT(NSTEP)
      GO TO 60
   50 IF(HINT.LE.0.)GO TO 20
      LSTEP=1
   60  RETURN
       END
       SUBROUTINE SINT(IPASS,NAM)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON XINT(10,10),DXINT(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YINT(10,10),DYINT(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
       HON2=HS/2.00
       IF(IPASS)1,1,3
    1  IF(NAM.GT.0)GO TO 15
       DO 4 I=1,K
       DO 4 J=1,K
       XINT(I,J)=XINTA1(I,J)+HS*DXINT(I,J)
    4  XINTA2(I,J)=DXINT(I,J)
      GO TO 10
   15  DO 16 I=1,K
       DO 16 J=1,K
       YINT(I,J)=YINTA1(I,J)+HS*DYINT(I,J)
   16   YINTA2(I,J)=DYINT(I,J)
      GO TO 10
    3   IF(NAM.GT.0)GO TO 25
      DO 7 I=1,K
       DO 7 J=1,K
       XINT(I,J)=XINTA1(I,J)+HON2*(XINTA2(I,J)+DXINT(I,J))
    7  XINTA1(I,J)=XINT(I,J)
      GO TO 10
   25 DO 18 I=1,K
       DO 18 J=1,K
       YINT(I,J)=YINTA1(I,J)+HON2*(YINTA2(I,J)+DYINT(I,J))
   18   YINTA1(I,J)=YINT(I,J)
   10  CONTINUE
       RETURN
       END
       SUBROUTINE LAMBDA
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      COMPLEX*16 E(10),C(10,10),B(10,10),FI(10,10),PHI(10,10),GG(10,10),
     1QQ(10,10),QFI(10,10),F(20),WW(150)
      COMMON E,C,B,FI,PHI,GG,QQ,QFI,F
      COMMON  A1(10,10),A2(10,10),A3(10,10),A4(10,10),D1(10,10),
```

```
     1D2(10,10),D3(10,10),D4(10,10),H(20),QD1(10,10),QD2(10,10)
     2,A11(10,10)
      COMMON  W(200),ZZ(10,10),EI(10),ER(10),G(10,10),Q(10,10)
      COMMON D(20,20),TIME,HS,HSTEP,EPS,U
       COMMON PFI(10,10),DFI(10,10),XINTA1(10,10),XINTA2(10,10)
     1,YFI(10,10),DYFI(10,10),YINTA1(10,10),YINTA2(10,10)
      COMMON K,M,MMK, IEIG,INDEX(10),JCOUNT(10)
     1,KCOUNT,IMETH, NT,MTWO
      DIMENSION HNO(10),HNI(10)
       NT1=NT-1
       HUSER=H(1)
       DO 10 I=1,NT1
       HSTAB=H(I+1)-H(I)
       IF(HSTAB.LT.HUSER)HUSER=HSTAB
10     CONTINUE
      WRITE(6,15)HUSER
15    FORMAT(2X,'MINIMUM USER STEPLENGTH INTERVAL IS ',E13.6)
       EPS=0.100
      IF(ER(1).EQ.0.D0)GO TO 16
       HMIN =(-2.D0*EPS)/ER(1)
      GO TO 18
16    HMIN=2000.D0
18     DO 100 I=1,K
       IF(ER(I).EQ.0.D0)GO TO 20
       HNO(I)=(-2.D0*EPS)/ER(I)
       GO TO 30
 20    HNO(I)=10.D0
 30    IF(EI(I))100,40,50
 40    HNI(I)=10.D0
       GO TO 70
50      IF(HNO(I).EQ.10.D0)GO TO 60
      OMB=(-8.D0*ER(I)*HNO(I))**0.25
      VH=(EI(I)*HNO(I))/OMB
      IF(VH.GE.0.2D0)GO TO 60
      HLAM=HNO(I)
      GO TO 90
 60    OM=-ER(I)/(EI(I)**4)
       PSI=8.D0*((2.D0*EPS)**4)
       EXPN=1.D0/3.D0
       HNI(I)=(OM*PSI)**EXPN
 70     HLAM=DMIN1(HNO(I),HNI(I))
 90    IF(HLAM.LT.HMIN)HMIN=HLAM
100    CONTINUE
       HSTEP=DMIN1(HMIN,HUSER)
       WRITE(6,105)HSTEP
105    FORMAT(2X,'N.I. STEPLENGTH FROM S/R LAMBDA IS ',E13.6)
       RETURN
       END
      SUBROUTINE OUTPUT (XX,L,N,X)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      DIMENSION DUMMY(20),XX(10,10),X(L,N)
      DO 2 I=1,L
      DO 2 J=1,N
2     X(I,J)=XX(I,J)
       DO 10 I=1,L
       WRITE(6,5)(X(I,J),J=1,N)
5     FORMAT(1H ,10(F9.2,2X))
 10    CONTINUE
       WRITE(6,15)
 15    FORMAT(1A,100(1H.))
       RETURN
       END
```

APPENDIX IV

LIST OF VARIABLES USED IN SOFTWARE PACKAGE

A1, D1 A2, D2 A3, D3 A4, D4	This family of real matrices form the analogue and digital pairs which describe the partitioning scheme being tested. The row and column dimensions in the calling program are 10 for each matrix. The actual size of each matrix is defined by the user for his problem.
A11	Stores A1 which could be lost during execution of the IMSL eigenvalue routine EIGRF.
M	Number of first order d.e.'s in the system, $\dot{x} = Ax$, once reduced. ($M \geq K+1$).
K	Actual row or column size of A1, D1 ($K \leq 10$).
KK	= K x K; number of elements in A1, D1.
KMK	= K x (M-K) number of elements in A2, D2, A3, D3 ($M-K \leq 10$).
MKMK	= (M-K) x (M-K) number of elements in A4, D4.
MTWO	2M ($2M \leq 20$).
H	Real vector of length NT containing the user values of step-length for testing the partitioning scheme.
NT	Length of the vector H.
EL	First element of H; defined when the required step-lengths are at equal time intervals.
HINT	Time interval between successive step-lengths H_i and H_{i+1} when EL is defined.
IMETH	= 1 The eigenvalue method is used for computing, Φ, G, Q. = 2 The numerical integration method is used for Φ, G, Q. The user defined value for IMETH can be overwritten by the program depending on the value assigned to IEIG.
E	Complex vector of length K containing the eigenvalues λ_i of A1.
ER	Real vector defining the real part of E.
EI	Real vector defining imaginary part of E.
C	Complex matrix of order K x K of eigenvectors of A1.
B	Inverse matrix of C.
W	Working space, dimensioned at least (2 + K) x K, used in the library routine EIGRF.
JCOUNT	Vector, of length K, which is used to print the results of the stability analysis of G.

IST, IMST, INST — Counters for recording whether each eigenvalue of A1 has negative, zero or positive real part. The result defines the stability of G.

IEIG — Counter for accumulating the number of repeated eigenvalues λ_i. IEIG = 0 means the λ_i are distinct.

TIME — Set to H_i for i = 1,...,NT.

FI — Complex matrix defined by $(\exp(\lambda_1 h),\ldots,\exp(\lambda_k h))$.

QFI — Complex matrix defined by $\left(\exp\frac{(\lambda_1 h)-1}{\lambda_1},\ldots,\exp\frac{(\lambda_k h)-1}{\lambda_k}\right)$

PHI — Complex matrix defining (i) CxFI when computing G(h) and (ii) CxQFI when computing Q(h).

GG — PHI x B for (i) above.

QQ — PHI x B for (ii) above.
GG and QQ are complex matrices. It is expected the imaginary part of these matrices will be less than 10^{-3}.

*NOTE** — *FI,QFI,PHI,GG and QQ are all defined in subroutine FUND for a range of step-lengths defined by H. The order of each matrix is K, while the row and column dimension in the calling program is 10.*

G — Real matrix of order K; = GG for IMETH = 1
= $\Phi(h)$ for IMETH = 2

Q — Real matrix of order K; = QQ for IMETH = 1
= $\int_0^h \Phi(\tau)d\tau$ for IMETH = 2
Both G and Q are defined for a range of step-lengths H.

NAM — = -1; G being computed in numerical integration routines.
= +1; Q being computed in numerical integration routines.

IPASS — = 0 Set in Main for initialisation in integration routine SINT.
= -1 Set in subroutine NUMERI for operation of the "predict" step of SINT.
= +1 Set in S/R NUMERI for operation of the "correct" step in SINT.

TAU — Keeps tally of time from 0 to H_i for numerical integration.

EPS — Describes the accuracy of the numerical solution for $\Phi(h)$.

HNO, HNI — Real vectors of length K, defining values of N.I step-length as functions of λ_j, for j = 1, ..., K.

HUSER — Minimum step-length interval between elements of the vector H.

HMIN — Minimum of HNO(j), HNI(j) over the j eigenvalues, λ_j.

HSTEP	Minimum of HUSER, HMIN, and defined as the numerical integration step-length in subroutine LAMBDA.
HS	N.I step-length based on HSTEP, but modified to suit the interval (H_i, H_{i+1}) for i = 1, ..., (NT-1).
NSTEP	Integral number of steps of length HS in the interval (H_i, H_{i+1}).
XINT	Variable $x(t_{n+1})$, computed in numerical integration routine SINT, where $x(t_{n+1}) = x(t_n) + h\,\dot{x}(t_n)$ for "predict" step and $x(t_{n+1}) = x(t_n) + h/2(\dot{x}(t_n) + \dot{x}(t_{n-1}))$ for "correct" step. Used for computing G(h).
YINT	Analagous to XINT, but used for computing Q(h).
DXINT, DYINT	Define $\dot{x}(t_n)$ in SINT, for G(h), Q(h) respectively.
XINTA2, YINTA2	Hold previously computed values for DXINT, and DYINT respectively from the "predict" step in SINT.
XINTA1, YINTA1	Hold previously computed values for XINT and YINT respectively from the "correct" step of SINT.
PFI	Defined in NUMERI as $\Phi(t_{n+1})$, and is equivalent to XINT from SINT.
YFI	Defined in NUMERI as $G(t_{n+1})$, and is equivalent to YINT from SINT.
DFI	Defined in NUMERI as $\dot{\Phi}(t_n) = A1.\ \Phi(t_n)$, and is equivalent to DXINT.
DYFI	Defined in NUMERI as $\dot{Q}(t_n) = G(t_n)$, and is equivalent to DYINT.
ZZ	K x K working matrix
QD1	Real K x K matrix defined by Q x D1.
QD2	Real K x (M-K) matrix defined by Q x (A2 + D2).
D	Real 2M x 2M matrix which is a function of matrices G, Q, QD1, QD2, A3, D3, A4, D4. Defined in Section 3.2.
F	Complex vector defining 2M eigenvalues, μ_i, of matrix D.

KCOUNT	Counter defining the number of μ_i, with modulus greater than unity.
INDEX	Vector for storing the value of those μ_i, with modulus greater than unity.

APPENDIX V

SOLUTION TO THE EQUATION $\dot{\underline{x}}(t) = B\underline{x}(t) + C\underline{u}(t)$

Consider the general equation

$$\dot{\underline{x}}(t) = B\underline{x}(t) + C\underline{u}(t) \tag{V.1}$$

and let X(t) be a fundamental matrix of the n x n system

$$\dot{\underline{x}}(t) = B\underline{x}(t),$$

i.e. X is a matrix such that its n columns consist of n linearly independent solutions to the system. Then, by definition, X(t) satisfies the matrix differential equation

$$\frac{dX(t)}{dt} = BX(t) \tag{V.2}$$

A fundamental matrix is always non-singular (by definition), and is uniquely determined for a given set of initial conditions(ref.4).

We must prove that the solution for (V.1) is

$$\underline{x}(t) = X(t)\, X^{-1}(0)\, \underline{x}_0 + \int_0^t X(t)\, X^{-1}(\xi)\, C\underline{u}\,(\xi)\, d\xi$$

$$= \underbrace{X(t)\, X^{-1}(0)\, \underline{x}_0}_{\text{complementary solution}} + \underbrace{\underline{y}(t)}_{\text{particular solution}} \quad , \text{ say}$$

with

$$\underline{y}(0) = \underline{0}.$$

To establish this, we need only prove that the particular solution, $\underline{y}(t)$, satisfies equation (V.1), since the complementary solution is well established(ref.4).

That is, we have to show

$$\frac{d\underline{y}(t)}{dt} = B\,\underline{y}(t) + C\underline{u}(t) \tag{V.3}$$

Put

$$v(t) = X(t)\, Z(t)$$

Then

$$\dot{\underline{y}} = \dot{X}\underline{Z} + X\dot{\underline{Z}}$$
$$= BX\underline{Z} + X\dot{\underline{Z}} \qquad \text{from equation (V.2)}$$
$$= B\underline{y} + X\dot{\underline{Z}}$$

We have from equation (V.3)

$$\dot{\underline{y}} = B\underline{y} + C\underline{u}$$

$\therefore$ we must have

$$X\dot{\underline{Z}} = C\underline{u}$$

i.e.

$$\dot{\underline{Z}} = X^{-1}(t)\, C\underline{u}(t)$$

$\therefore$

$$\underline{Z}(t) = \int_o^t X^{-1}(\xi)\, C\underline{u}(\xi)\, d\xi$$

and

$$\underline{y}(t) = \int_o^t X(t)\, X^{-1}(\xi)\, C\underline{u}(\xi)\, d\xi$$

Choose $X(t)$ such that $X(0) = I$. By definition, since $X(t)$ is unique,

$$X(t) = e^{Bt}$$

Also, by definition,

$$X^{-1}(\xi) = e^{-B\xi} = X(-\xi)$$

and hence

$$X(t)\, X^{-1}(\xi) = e^{B(t-\xi)} = X(t - \xi)$$

Thus

$$\underline{y}(t) = \int_o^t X(t - \xi)\, C\underline{u}(\xi)\, d\xi$$

and

$$\underline{x}(t) = X(t)\,\underline{x}_0 + \int_o^t X(t-\xi)\,C\underline{u}(\xi)\,d\xi$$

If $\underline{u}(\xi) = \underline{u}_N$ = constant, say, and $t = h$

$$\underline{x}(h) = X(h)\,\underline{x}_0 + \int_o^h X(h-\xi)\,d\xi\;C\underline{u}_N$$

Replacing $h - \xi$ by τ, it can be shown that

$$\underline{x}(h) = X(h)\,\underline{x}_0 + \int_o^h X(\tau)\,d\tau\;C\underline{u}_N$$

However, any fundamental matrix, $\Phi(t)$, for the system, can be written as the product of the unique matrix $X(t)$ and a non-singular constant matrix C.

$$\text{i.e. } \Phi(t) = X(t)C$$

Thus

$$C = \Phi^{-1}(0)$$

and

$$X(t) = \Phi(t)\,\Phi^{-1}(0)$$

giving

$$\underline{x}(h) = \Phi(h)\,\Phi^{-1}(0)\,\underline{x}_0 + \int_o^h \Phi(\tau)\,\Phi^{-1}(0)\,d\tau\;C\underline{u}_N$$

APPENDIX VI

RESULTS OF APPLICATION 5.1

```
INPUT DATA
..........

K= 4  M= 7
A1 MATRIX-( 4 X 4)
       0.0          1.00          0.0          0.0
       0.0         -5.80        11.40       -10.64
       0.0           0.0       -43.50          0.0
       0.0          1.00        -0.60          0.0
.....
D1 MATRIX-( 4 X 4)
       0.0           0.0          0.0          0.0
    -92.80           0.0          0.0          0.0
   -459.10           0.0          0.0          0.0
      4.60           0.0          0.0          0.0
.....
A2 MATRIX-( 4 X 3)
       0.0           0.0          0.0
       0.0           0.0          0.0
       0.0           0.0          0.0
       0.0           0.0          0.0
.....
D2 MATRIX-( 4 X 3)
       0.0           0.0          0.0
    -92.80           0.0          0.0
   -459.10           0.0          0.0
      4.60           0.0          0.0
.....
A3 MATRIX-( 3 X 4)
       0.0           0.0          0.0          0.0
       0.0           0.0          0.0          0.0
       0.0           0.0          0.0          0.0
.....
D3 MATRIX-( 3 X 4)
      -6.30           0.0          0.0          0.0
      -3.00           0.0          0.0          0.0
       1.00           0.0          0.0          0.0
.....
A4 MATRIX-( 3 X 3)
       0.0           0.0          0.0
       0.0           0.0          0.0
       0.0           0.0          0.0
.....
D4 MATRIX-( 3 X 3)
     -13.80          6.30       -62.50
      -3.00           0.0          0.0
       1.00         -1.00          0.0
.....
THE  6 STEPLENGTHS ARE-
0.0380  0.0390  0.0400  0.0410  0.0420  0.0430

IMETH=2
```

```
ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

THE EIGENVALUE ROUTINE PERFORMED WELL

        EIGENVALUES OF A1 (COMPLEX ELEMENTS)
        ------------------------------------
( 0.0       , 0.0       )  (-0.2900D+01, 0.1493D+01)  (-0.2900D+01,-0.1493D+01)  (-0.4350D+02, 0.0       )
-------------------------------------------------------------------------------------------------------------

 1 OF THE ZEROS OF A1 HAVE ZERO REAL PART.G MATRIX WILL BE MARGINALLY STABLE

THE EIGENVALUES OF A1 ARE DISTINCT

MINIMUM USER STEPLENGTH INTERVAL IS  0.100000D-02
M.I. STEPLENGTH FROM S/R LAMBDA IS  0.100000D-02

H=0.0380 NUMERICAL INTEGRATION STEPLENGTH IS 0.001000

        G MATRIX

 ROW  1  0.1000D+01  0.3402D-01  0.4724D-02 -0.7139D-02

 ROW  2  0.0         0.7956D+00  0.1866D+00 -0.3619D+00

 ROW  3  0.0         0.0         0.1916D+00  0.0

 ROW  4  0.0         0.3402D-01 -0.6427D-02  0.9929D+00
-------------------------------------------------------------------------------------------------------------
        C MATRIX

 ROW  1  0.3800D-01  0.6709D-03  0.6856D-04 -0.9214D-04

 ROW  2  0.0         0.3402D-01  0.4721D-02 -0.7138D-02

 ROW  3  0.0         0.0         0.1859D-01  0.0

 ROW  4  0.0         0.6709D-03 -0.1991D-03  0.3791D-01
-------------------------------------------------------------------------------------------------------------
        D MATRIX

 ROW  1  0.1000D+01  0.3402D-01  0.4724D-02 -0.7139D-02 -0.9416D-01  0.0         0.0         0.1000D+01  0.0         0.0
         0.0         0.0         0.0         0.0

 ROW  2  0.0         0.7956D+00  0.1866D+00 -0.3619D+00 -0.5357D+01  0.0         0.0         0.0         0.1000D+01  0.0
         0.0         0.0         0.0         0.0

 ROW  3  0.0         0.0         0.1916D+00  0.0        -0.8536D+01  0.0         0.0         0.0         0.0         0.1000D+01
         0.0         0.0         0.0         0.0

 ROW  4  0.0         0.3402D-01 -0.6427D-02  0.9929D+00  0.2035D+00  0.0         0.0         0.0         0.0         0.0
         0.1000D+01  0.0         0.0         0.0

 ROW  5 -0.2394D+00  0.0         0.0         0.0         0.4756D+00  0.2394D+00 -0.2375D+01  0.0         0.0         0.0
         0.0         0.1000D+01  0.0         0.0

 ROW  6 -0.1140D+00  0.0         0.0         0.0        -0.1140D+00  0.1000D+01  0.0         0.0         0.0         0.0
         0.0         0.0         0.1000D+01  0.0
```

```
 ROW  7  0.3800D-01  0.0          0.0          0.0          0.3800D-01 -0.3800D-01  0.1000D+01  0.0          0.0          0.0
         0.0         0.0          0.0          0.1000D+01

 ROW  8 -0.9416D-01  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW  9 -0.5357D+01  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 10 -0.8536D+01  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 11  0.2035D+00  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 12  0.0         0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 13  0.0         0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 14  0.0         0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0
-------------------------------------------------------------------------------------------------------------------------------
 ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
        -----------------------------------
( 0.8602D+00, 0.4999D+00)  ( 0.8602D+00,-0.4999D+00)  ( 0.7333D+00, 0.0       )  ( 0.9911D-02, 0.3085D-01)

( 0.9911D-02,-0.3085D-01)  ( 0.9911D+00, 0.2949D-01)  ( 0.9911D+00,-0.2949D-01)  ( 0.1000D+01, 0.0       )

( 0.7280D-14, 0.0       )  ( 0.4509D-16, 0.0       )  (-0.6334D-17, 0.0       )  ( 0.0       , 0.0       )

( 0.0       , 0.0       )  ( 0.0       , 0.0       )  (
 ALL THE EIGENVALUES OF D HAVE MODULUS LESS THAN OR EQUAL TO UNITY FOR H=0.0380 SCHEME IS STABLE FOR CHOICE OF H
*******************************************************************************************************************************

 H=0.0390 NUMERICAL INTEGRATION STEPLENGTH IS 0.000500

        G MATRIX

 ROW  1  0.1000D+01  0.3481D-01  0.4911D-02 -0.7505D-02

 ROW  2  0.0         0.7906D+00  0.1877D+00 -0.3704D+00

 ROW  3  0.0         0.0         0.1834D+00  0.0

 ROW  4  0.0         0.3481D-01 -0.6352D-02  0.9925D+00
-------------------------------------------------------------------------------------------------------------------------------
        Q MATRIX

 ROW  1  0.3900D-01  0.7053D-03  0.7337D-04 -0.9946D-04

 ROW  2  0.0         0.3481D-01  0.4908D-02 -0.7504D-02

 ROW  3  0.0         0.0         0.1878D-01  0.0

 ROW  4  0.0         0.7053D-03 -0.2055D-03  0.3890D-01
-------------------------------------------------------------------------------------------------------------------------------
        D MATRIX
```

```
ROW  1  0.1000D+01  0.3481D-01  0.4911D-02 -0.7505D-02 -0.9960D-01  0.0          0.0          0.1000D+01  0.0          0.0
        0.0         0.0         0.0         0.0

ROW  2  0.0         0.7906D+00  0.1877D+00 -0.3704D+00 -0.5518D+01  0.0          0.0          0.0         0.1000D+01  0.0
        0.0         0.0         0.0         0.0

ROW  3  0.0         0.0         0.1834D+00  0.0         -0.8622D+01  0.0          0.0          0.0         0.0         0.1000D+01
        0.0         0.0         0.0         0.0

ROW  4  0.0         0.3481D-01 -0.6352D-02  0.9925D+00  0.2078D+00  0.0          0.0          0.0         0.0         0.0
        0.1000D+01  0.0         0.0         0.0

ROW  5 -0.2457D+00  0.0         0.0         0.0         0.4618D+00  0.2457D+00 -0.2437D+01  0.0         0.0         0.0
        0.0         0.1000D+01  0.0         0.0

ROW  6 -0.1170D+00  0.0         0.0         0.0         -0.1170D+00  0.1000D+01  0.0          0.0         0.0         0.0
        0.0         0.0         0.1000D+01  0.0

ROW  7  0.3900D-01  0.0         0.0         0.0         0.3900D-01 -0.3900D-01  0.1000D+01  0.0         0.0         0.0
        0.0         0.0         0.0         0.1000D+01

ROW  8 -0.9960D-01  0.0         0.0         0.0         0.0          0.0          0.0          0.0         0.0         0.0
        0.0         0.0         0.0         0.0

ROW  9 -0.5518D+01  0.0         0.0         0.0         0.0          0.0          0.0          0.0         0.0         0.0
        0.0         0.0         0.0         0.0

ROW 10 -0.8622D+01  0.0         0.0         0.0         0.0          0.0          0.0          0.0         0.0         0.0
        0.0         0.0         0.0         0.0

ROW 11  0.2078D+00  0.0         0.0         0.0         0.0          0.0          0.0          0.0         0.0         0.0
        0.0         0.0         0.0         0.0

ROW 12  0.0         0.0         0.0         0.0         0.0          0.0          0.0          0.0         0.0         0.0
        0.0         0.0         0.0         0.0

ROW 13  0.0         0.0         0.0         0.0         0.0          0.0          0.0          0.0         0.0         0.0
        0.0         0.0         0.0         0.0

ROW 14  0.0         0.0         0.0         0.0         0.0          0.0          0.0          0.0         0.0         0.0
        0.0         0.0         0.0         0.0
-------------------------------------------------------------------------------------------------------------------------

ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
        -----------------------------------

( 0.8561D+00, 0.5105D+00)  ( 0.8561D+00,-0.5105D+00)  ( 0.3402D-02, 0.3785D-01)  ( 0.3402D-02,-0.3785D-01)

( 0.9913D+00, 0.3016D-01)  ( 0.9913D+00,-0.3016D-01)  ( 0.1000D+01, 0.0       )  ( 0.7267D+00, 0.0       )

( 0.1205D-13, 0.0       )  (-0.4962D-16, 0.0       )  ( 0.3276D-16, 0.0       )  ( 0.0       , 0.0       )

( 0.0       , 0.0       )  ( 0.0       , 0.0       )  (
ALL THE EIGENVALUES OF D HAVE MODULUS LESS THAN OR EQUAL TO UNITY FOR H=0.0390 SCHEME IS STABLE FOR CHOICE OF H
*************************************************************************************************************

H=0.0400 NUMERICAL INTEGRATION STEPLENGTH IS 0.000500

        G MATRIX

 ROW  1  0.1000D+01  0.3560D-01  0.5099D-02 -0.7880D-02
```

```
 ROW  2  0.0           0.7857D+00  0.1887D+00 -0.3788D+00

 ROW  3  0.0           0.0         0.1756D+00  0.0

 ROW  4  0.0           0.3560D-01 -0.6272D-02  0.9921D+00
-------------------------------------------------------------------------------------------------------------------
        C MATRIX

 ROW  1  0.4000D-01    0.7405D-03  0.7838D-04 -0.1072D-03

 ROW  2  0.0           0.3560D-01  0.5097D-02 -0.7879D-02

 ROW  3  0.0           0.0         0.1896D-01  0.0

 ROW  4  0.0           0.7405D-03 -0.2118D-03  0.3989D-01
-------------------------------------------------------------------------------------------------------------------
        D MATRIX

 ROW  1  0.1000D+01    0.3560D-01  0.5099D-02 -0.7880D-02 -0.1052D+00  0.0          0.0          0.1000D+01  0.0          0.0
         0.0           0.0         0.0         0.0

 ROW  2  0.0           0.7857D+00  0.1887D+00 -0.3788D+00 -0.5680D+01  0.0          0.0          0.0         0.1000D+01  0.0
         0.0           0.0         0.0         0.0

 ROW  3  0.0           0.0         0.1756D+00  0.0        -0.8705D+01  0.0          0.0          0.0         0.0         0.1000D+01
         0.0           0.0         0.0         0.0

 ROW  4  0.0           0.3560D-01 -0.6272D-02  0.9921D+00  0.2120D+00  0.0          0.0          0.0         0.0         0.0
         0.1000D+01    0.0         0.0         0.0

 ROW  5 -0.2520D+00    0.0         0.0         0.0         0.4480D+00  0.2520D+00  -0.2500D+01   0.0         0.0         0.0
         0.0           0.1000D+01  0.0         0.0

 ROW  6 -0.1200D+00    0.0         0.0         0.0        -0.1200D+00  0.1000D+01   0.0          0.0         0.0         0.0
         0.0           0.0         0.1000D+01  0.0

 ROW  7  0.4000D-01    0.0         0.0         0.0         0.4000D-01 -0.4000D-01   0.1000D+01   0.0         0.0         0.0
         0.0           0.0         0.0         0.1000D+01

 ROW  8 -0.1052D+00    0.0         0.0         0.0         0.0         0.0          0.0          0.0         0.0         0.0
         0.0           0.0         0.0         0.0

 ROW  9 -0.5680D+01    0.0         0.0         0.0         0.0         0.0          0.0          0.0         0.0         0.0
         0.0           0.0         0.0         0.0

 ROW 10 -0.8705D+01    0.0         0.0         0.0         0.0         0.0          0.0          0.0         0.0         0.0
         0.0           0.0         0.0         0.0

 ROW 11  0.2120D+00    0.0         0.0         0.0         0.0         0.0          0.0          0.0         0.0         0.0
         0.0           0.0         0.0         0.0

 ROW 12  0.0           0.0         0.0         0.0         0.0         0.0          0.0          0.0         0.0         0.0
         0.0           0.0         0.0         0.0

 ROW 13  0.0           0.0         0.0         0.0         0.0         0.0          0.0          0.0         0.0         0.0
         0.0           0.0         0.0         0.0

 ROW 14  0.0           0.0         0.0         0.0         0.0         0.0          0.0          0.0         0.0         0.0
         0.0           0.0         0.0         0.0
-------------------------------------------------------------------------------------------------------------------
 ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0
```

EIGENVALUES OF D (COMPLEX ELEMENTS)

(0.8520D+00, 0.5210D+00) (0.8520D+00,-0.5210D+00) (-0.2921D-02, 0.4282D-01) (-0.2921D-02,-0.4282D-01)

(0.9915D+00, 0.3081D-01) (0.9915D+00,-0.3081D-01) (0.1000D+01, 0.0) (0.7201D+00, 0.0)

(0.1076D-13, 0.0) (0.5499D-16, 0.0) (-0.1332D-16, 0.0) (0.0 , 0.0)

(0.0 , 0.0) (0.0 , 0.0) (

ALL THE EIGENVALUES OF D HAVE MODULUS LESS THAN OR EQUAL TO UNITY FOR H=0.0400 SCHEME IS STABLE FOR CHOICE OF H

****

H=0.0410 NUMERICAL INTEGRATION STEPLENGTH IS 0.000500

G MATRIX

ROW 1	0.1000D+01	0.3638D-01	0.5288D-02	-0.8262D-02
ROW 2	0.0	0.7807D+00	0.1897D+00	-0.3871D+00
ROW 3	0.0	0.0	0.1681D+00	0.0
ROW 4	0.0	0.3638D-01	-0.6185D-02	0.9917D+00

C MATRIX

ROW 1	0.4100D-01	0.7765D-03	0.8357D-04	-0.1152D-03
ROW 2	0.0	0.3638D-01	0.5286D-02	-0.8262D-02
ROW 3	0.0	0.0	0.1913D-01	0.0
ROW 4	0.0	0.7765D-03	-0.2181D-03	0.4088D-01

D MATRIX

ROW 1	0.1000D+01	0.3638D-01	0.5288D-02	-0.8262D-02	-0.1110D+00	0.0	0.0	0.1000D+01	0.0	0.0
	0.0	0.0	0.0	0.0						
ROW 2	0.0	0.7807D+00	0.1897D+00	-0.3871D+00	-0.5841D+01	0.0	0.0	0.0	0.1000D+01	0.0
	0.0	0.0	0.0	0.0						
ROW 3	0.0	0.0	0.1681D+00	0.0	-0.8784D+01	0.0	0.0	0.0	0.0	0.1000D+01
	0.0	0.0	0.0	0.0						
ROW 4	0.0	0.3638D-01	-0.6185D-02	0.9917D+00	0.2161D+00	0.0	0.0	0.0	0.0	0.0
	0.1000D+01	0.0	0.0	0.0						
ROW 5	-0.2583D+00	0.0	0.0	0.0	0.4342D+00	0.2583D+00	-0.2562D+01	0.0	0.0	0.0
	0.0	0.1000D+01	0.0	0.0						
ROW 6	-0.1230D+00	0.0	0.0	0.0	-0.1230D+00	0.1000D+01	0.0	0.0	0.0	0.0
	0.0	0.0	0.1000D+01	0.0						
ROW 7	0.4100D-01	0.0	0.0	0.0	0.4100D-01	-0.4100D-01	0.1000D+01	0.0	0.0	0.0
	0.0	0.0	0.0	0.1000D+01						
ROW 8	-0.1110D+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0						

```
 ROW  9 -0.5841D+01  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 10 -0.8784D+01  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 11  0.2161D+00  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 12  0.0         0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 13  0.0         0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0

 ROW 14  0.0         0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
         0.0         0.0          0.0          0.0
-------------------------------------------------------------------------------------------------------------------------------
 ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
        -----------------------------------
(-0.9064D-02, 0.4647D-01)  (-0.9064D-02,-0.4647D-01)  ( 0.8479D+00, 0.5314D+00)  ( 0.8479D+00,-0.5314D+00)

( 0.7136D+00, 0.0       )  ( 0.9918D+00, 0.3144D-01)  ( 0.9918D+00,-0.3144D-01)  ( 0.1000D+01, 0.0       )

( 0.6712D-14, 0.0       )  ( 0.5825D-16, 0.0       )  (-0.5613D-17, 0.0       )  ( 0.0       , 0.0       )

( 0.0       , 0.0       )  ( 0.0       , 0.0       )  (
FOR H=0.0410,  2 OF THE EIGENVALUES OF D HAVE MODULUS GREATER THAN UNITY.THEREFORE CHOICE IS UNSTABLE FOR THIS H
( 0.8479D+00, 0.5314D+00)  ( 0.8479D+00,-0.5314D+00)  (
*******************************************************************************************************************************

 H=0.0420 NUMERICAL INTEGRATION STEPLENGTH IS 0.000500

        G MATRIX

 ROW  1  0.1000D+01  0.3716D-01  0.5479D-02 -0.8654D-02

 ROW  2  0.0         0.7758D+00  0.1905D+00 -0.3954D+00

 ROW  3  0.0         0.0         0.1610D+00  0.0

 ROW  4  0.0         0.3716D-01 -0.6094D-02  0.9913D+00
-------------------------------------------------------------------------------------------------------------------------------
        C MATRIX

 ROW  1  0.4200D-01  0.8133D-03  0.8896D-04 -0.1237D-03

 ROW  2  0.0         0.3716D-01  0.5476D-02 -0.8653D-02

 ROW  3  0.0         0.0         0.1930D-01  0.0

 ROW  4  0.0         0.8133D-03 -0.2242D-03  0.4188D-01
-------------------------------------------------------------------------------------------------------------------------------
        D MATRIX

 ROW  1  0.1000D+01  0.3716D-01  0.5479D-02 -0.8654D-02 -0.1169D+00  0.0          0.0          0.1000D+01  0.0          0.0
         0.0         0.0          0.0          0.0

 ROW  2  0.0         0.7758D+00  0.1905D+00 -0.3954D+00 -0.6002D+01  0.0          0.0          0.0          0.1000D+01  0.0
```

```
          0.0           0.0           0.0           0.0

 ROW  3   0.0           0.0           0.1610D+00    0.0          -0.8859D+01   0.0           0.0           0.0           0.0           0.1000D+01
          0.0           0.0           0.0           0.0

 ROW  4   0.0           0.3716D-01   -0.6094D-02    0.9913D+00    0.2201D+00   0.0           0.0           0.0           0.0           0.0
          0.1000D+01    0.0           0.0           0.0

 ROW  5  -0.2646D+00    0.0           0.0           0.0           0.4204D+00   0.2646D+00   -0.2625D+01    0.0           0.0           0.0
          0.0           0.1000D+01    0.0           0.0

 ROW  6  -0.1260D+00    0.0           0.0           0.0          -0.1260D+00   0.1000D+01    0.0           0.0           0.0           0.0
          0.0           0.0           0.1000D+01    0.0

 ROW  7   0.4200D-01    0.0           0.0           0.0           0.4200D-01  -0.4200D-01    0.1000D+01    0.0           0.0           0.0
          0.0           0.0           0.0           0.1000D+01

 ROW  8  -0.1169D+00    0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0
          0.0           0.0           0.0           0.0

 ROW  9  -0.6002D+01    0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0
          0.0           0.0           0.0           0.0

 ROW 10  -0.8859D+01    0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0
          0.0           0.0           0.0           0.0

 ROW 11   0.2201D+00    0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0
          0.0           0.0           0.0           0.0

 ROW 12   0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0
          0.0           0.0           0.0           0.0

 ROW 13   0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0
          0.0           0.0           0.0           0.0

 ROW 14   0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0           0.0
          0.0           0.0           0.0           0.0
-------------------------------------------------------------------------------------------------------------------------------------------
ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
        -----------------------------------

( 0.8437D+00, 0.5416D+00)  ( 0.8437D+00,-0.5416D+00)  (-0.1503D-01, 0.4915D-01)  (-0.1503D-01,-0.4915D-01)

( 0.9921D+00, 0.3206D-01)  ( 0.9921D+00,-0.3206D-01)  ( 0.1000D+01, 0.0       )  ( 0.7070D+00, 0.0       )

(-0.5211D-15, 0.0       )  ( 0.3170D-16, 0.5643D-16)  ( 0.3170D-16,-0.5643D-16)  ( 0.0       , 0.0       )

( 0.0       , 0.0       )  ( 0.0       , 0.0       )  (
FOR H=0.0420,  2 OF THE EIGENVALUES OF D HAVE MODULUS GREATER THAN UNITY.THEREFORE CHOICE IS UNSTABLE FOR THIS H
( 0.8437D+00, 0.5416D+00)  ( 0.8437D+00,-0.5416D+00)  (
*******************************************************************************************************************************************

H=0.0430 NUMERICAL INTEGRATION STEPLENGTH IS 0.000500

        G MATRIX

 ROW  1  0.1000D+01  0.3793D-01  0.5669D-02 -0.9053D-02

 ROW  2  0.0         0.7709D+00  0.1912D+00 -0.4036D+00
```

```
 ROW  3  0.0           0.0           0.1541D+00  0.0

 ROW  4  0.0           0.3793D-01 -0.5998D-02  0.9909D+00
-------------------------------------------------------------------------------------------------------------------
        C MATRIX

 ROW  1  0.4300D-01    0.8508D-03    0.9453D-04 -0.1325D-03

 ROW  2  0.0           0.3793D-01    0.5667D-02 -0.9053D-02

 ROW  3  0.0           0.0           0.1945D-01  0.0

 ROW  4  0.0           0.8508D-03 -0.2302D-03   0.4287D-01
-------------------------------------------------------------------------------------------------------------------
        D MATRIX

 ROW  1  0.1000D+01    0.3793D-01    0.5669D-02 -0.9053D-02 -0.1230D+00  0.0          0.0          0.1000D+01  0.0          0.0
         0.0           0.0           0.0         0.0

 ROW  2  0.0           0.7709D+00    0.1912D+00 -0.4036D+00 -0.6163D+01  0.0          0.0          0.0         0.1000D+01   0.0
         0.0           0.0           0.0         0.0

 ROW  3  0.0           0.0           0.1541D+00  0.0        -0.8931D+01  0.0          0.0          0.0         0.0          0.1000D+01
         0.0           0.0           0.0         0.0

 ROW  4  0.0           0.3793D-01 -0.5998D-02   0.9909D+00  0.2239D+00  0.0          0.0          0.0         0.0          0.0
         0.1000D+01    0.0           0.0         0.0

 ROW  5 -0.2709D+00    0.0           0.0         0.0         0.4066D+00  0.2709D+00 -0.2688D+01   0.0         0.0          0.0
         0.0           0.1000D+01    0.0         0.0

 ROW  6 -0.1290D+00    0.0           0.0         0.0        -0.1290D+00  0.1000D+01   0.0          0.0         0.0          0.0
         0.0           0.0           0.1000D+01  0.0

 ROW  7  0.4300D-01    0.0           0.0         0.0         0.4300D-01 -0.4300D-01   0.1000D+01   0.0         0.0          0.0
         0.0           0.0           0.0         0.1000D+01

 ROW  8 -0.1230D+00    0.0           0.0         0.0         0.0         0.0          0.0          0.0         0.0          0.0
         0.0           0.0           0.0         0.0

 ROW  9 -0.6163D+01    0.0           0.0         0.0         0.0         0.0          0.0          0.0         0.0          0.0
         0.0           0.0           0.0         0.0

 ROW 10 -0.8931D+01    0.0           0.0         0.0         0.0         0.0          0.0          0.0         0.0          0.0
         0.0           0.0           0.0         0.0

 ROW 11  0.2239D+00    0.0           0.0         0.0         0.0         0.0          0.0          0.0         0.0          0.0
         0.0           0.0           0.0         0.0

 ROW 12  0.0           0.0           0.0         0.0         0.0         0.0          0.0          0.0         0.0          0.0
         0.0           0.0           0.0         0.0

 ROW 13  0.0           0.0           0.0         0.0         0.0         0.0          0.0          0.0         0.0          0.0
         0.0           0.0           0.0         0.0

 ROW 14  0.0           0.0           0.0         0.0         0.0         0.0          0.0          0.0         0.0          0.0
         0.0           0.0           0.0         0.0
-------------------------------------------------------------------------------------------------------------------
ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
```

```
 ----------------------------------
( 0.8395D+00, 0.5518D+00)  ( 0.8395D+00,-0.5518D+00)  (-0.2083D-01, 0.5105D-01)  (-0.2083D-01,-0.5105D-01)

( 0.9924D+00, 0.3266D-01)  ( 0.9924D+00,-0.3266D-01)  ( 0.1000D+01, 0.0       )  ( 0.7005D+00, 0.0       )

( 0.2729D-14, 0.0       )  ( 0.5494D-16, 0.0       )  (-0.1522D-16, 0.0       )  ( 0.0       , 0.0       )

( 0.0       , 0.0       )  ( 0.0       , 0.0       )  (
FOR H=0.0430,  2 OF THE EIGENVALUES OF D MATRIX MODULUS GREATER THAN UNITY.THEREFORE CHOICE IS UNSTABLE FOR THIS H
( 0.8395D+00, 0.5518D+00)  ( 0.8395D+00,-0.5518D+00)  (
**********************************************************************************************************************
```

APPENDIX VII

RESULTS OF APPLICATION 5.2

```
INPUT DATA
..........

K= 7  M= 8
A1 MATRIX-( 7 X 7)
      1.00        0.0         0.0         0.0         0.0         0.0
      0.0         0.0         0.0         0.0         0.0         0.0
      0.0        19.50       -0.50        0.0         0.0         0.0
      0.0         0.0         0.0     -1000.00    -151.90      70.00
      0.0         0.0         0.0         0.0         1.00        0.0
      0.0         0.0         0.0         0.0         0.0         0.0
      0.0       975.00       75.00        0.0         0.0      -100.00
.....
D1 MATRIX-( 7 X 7)
     -0.60        0.0         0.0        -0.10        0.0         0.0
    -58.40        0.0         0.0        -61.60       0.0         0.0
      0.0         0.0         0.0         0.0         0.0         0.0
   1709.30        0.0         0.0        304.30       0.0         0.0
      0.0         0.0         0.0         0.0         0.0         0.0
   -193.80        0.0         0.0        -34.50       0.0         0.0
      0.0         0.0         0.0         0.0         0.0         0.0
.....
A2 MATRIX-( 7 X 1)
      0.0
      0.0
      0.0
      0.0
      0.0
      0.0
      0.0
.....
D2 MATRIX-( 7 X 1)
      0.0
      0.0
      0.0
      0.0
      0.0
      0.0
      0.0
.....
A3 MATRIX-( 1 X 7)
      0.0         0.0         0.0         0.0         0.0         0.0
      0.0
.....
D3 MATRIX-( 1 X 7)
      0.0         0.0         0.0         0.0         0.0         0.0
      0.0
.....
A4 MATRIX-( 1 X 1)
      0.0
.....
D4 MATRIX-( 1 X 1)
      0.0
.....
THE  5 STEPLENGTHS ARE-
0.0090   0.0100   0.0110   0.0120   0.0130

IRETH=2
```

ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE 0

THE EIGENVALUE ROUTINE PERFORMED WELL

EIGENVALUES OF A1 (COMPLEX ELEMENTS)

(0.0 , 0.0) (-0.1000D+04, 0.0) (-0.1000D+03, 0.0) (-0.5000D+00, 0.0)

(0.0 , 0.0) (0.0 , 0.0) (0.0 , 0.0) (

---

4 OF THE ZEROS OF A1 HAVE ZERO REAL PART.G MATRIX WILL BE MARGINALLY STABLE

THE EIGENVALUES OF A1 ARE REPEATED G AND Q WILL BE OBTAINED BY NUMERICAL INTEGRATION

MINIMUM USER STEPLENGTH INTERVAL IS 0.100000D-02
N.I. STEPLENGTH FROM S/R LAMBDA IS 0.200000D-03

H=0.0090 NUMERICAL INTEGRATION STEPLENGTH IS 0.000200

G MATRIX

ROW 1	0.1000D+01	0.9000D-02	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.1751D+00	0.9955D+00	0.0	0.0	0.0	0.0
ROW 4	0.0	0.3767D+00	0.2872D-01	0.1323D-03	-0.1519D+00	-0.1531D+00	0.3161D-01
ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.9000D-02	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
ROW 7	0.0	0.5830D+01	0.4439D+00	0.0	0.0	0.0	0.4066D+00

C MATRIX

ROW 1	0.9000D-02	0.4050D-04	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.9000D-02	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.7886D-03	0.8980D-02	0.0	0.0	0.0	0.0
ROW 4	0.0	0.1726D-02	0.1320D-03	0.1011D-02	-0.1214D-02	-0.1218D-02	0.3830D-03
ROW 5	0.0	0.0	0.0	0.0	0.9000D-02	0.4050D-04	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.9000D-02	0.0
ROW 7	0.0	0.3003D-01	0.2295D-02	0.0	0.0	0.0	0.5935D-02

D MATRIX

ROW 1	0.1000D+01	0.9000D-02	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01
	0.0	0.0	0.0	0.0	0.0	0.0				

```
ROW  3  0.0           0.1751D+00  0.9955D+00  0.0           0.0           0.0           0.0           0.0           0.0           0.0
        0.1000D+01    0.0         0.0         0.0           0.0           0.0

ROW  4  0.0           0.3767D+00  0.2872D-01  0.1323D-03   -0.1519D+00   -0.1531D+00   0.3161D-01    0.0           0.0           0.0
        0.0           0.1000D+01  0.0         0.0           0.0           0.0

ROW  5  0.0           0.0         0.0         0.0           0.1000D+01    0.9000D-02    0.0           0.0           0.0           0.0
        0.0           0.0         0.1000D+01  0.0           0.0           0.0

ROW  6  0.0           0.0         0.0         0.0           0.0           0.1000D+01    0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.1000D+01    0.0           0.0

ROW  7  0.0           0.5830D+01  0.4439D+00  0.0           0.0           0.0           0.4066D+00    0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.1000D+01    0.0

ROW  8  0.0           0.0         0.0         0.0           0.0           0.0           0.0           0.1000D+01    0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.1000D+01

ROW  9 -0.7765D-02    0.0         0.0        -0.3395D-02    0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0

ROW 10 -0.5256D+00    0.0         0.0        -0.5544D+00    0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0

ROW 11 -0.4605D-01    0.0         0.0        -0.4858D-01    0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0

ROW 12  0.1863D+01    0.0         0.0         0.2433D+00    0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0

ROW 13 -0.7849D-02    0.0         0.0        -0.1397D-02    0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0

ROW 14 -0.1744D+01    0.0         0.0        -0.3105D+00    0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0

ROW 15 -0.1754D+01    0.0         0.0        -0.1850D+01    0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0

ROW 16  0.0           0.0         0.0         0.0           0.0           0.0           0.0           0.0           0.0           0.0
        0.0           0.0         0.0         0.0           0.0           0.0
-------------------------------------------------------------------------------------------------------------------------------------
ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
        -----------------------------------
(-0.5818D+00, 0.0       ) ( 0.9069D+00, 0.3736D+00) ( 0.9069D+00,-0.3736D+00) ( 0.2177D+00, 0.0       )

( 0.9774D+00, 0.5099D-01) ( 0.9774D+00,-0.5099D-01) ( 0.9955D+00, 0.0       ) ( 0.9896D+00, 0.0       )

( 0.1269D-01, 0.0       ) (-0.9443D-16, 0.1509D-15) (-0.9443D-16,-0.1509D-15) ( 0.6387D-16, 0.0       )

(-0.3112D-17, 0.0       ) ( 0.6044D-19, 0.0       ) ( 0.1000D+01, 0.0       ) ( 0.0       , 0.0       )

ALL THE EIGENVALUES OF D HAVE MODULUS LESS THAN OR EQUAL TO UNITY FOR H=0.0090 SCHEME IS STABLE FOR CHOICE OF H
*************************************************************************************************************************************

H=0.0100 NUMERICAL INTEGRATION STEPLENGTH IS 0.000200

        G MATRIX
```

ROW 1	0.1000D+01	0.1000D-01	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.1945D+00	0.9950D+00	0.0	0.0	0.0	0.0
ROW 4	0.0	0.4067D+00	0.3096D-01	0.4906D-04	-0.1519D+00	-0.1533D+00	0.2861D-01
ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.1000D-01	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
ROW 7	0.0	0.6217D+01	0.4727D+00	0.0	0.0	0.0	0.3679D+00

C MATRIX

ROW 1	0.1000D-01	0.5000D-04	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.1000D-01	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.9734D-03	0.9975D-02	0.0	0.0	0.0	0.0
ROW 4	0.0	0.2118D-02	0.1619D-03	0.1011D-02	-0.1365D-02	-0.1372D-02	0.4130D-03
ROW 5	0.0	0.0	0.0	0.0	0.1000D-01	0.5000D-04	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D-01	0.0
ROW 7	0.0	0.3606D-01	0.2754D-02	0.0	0.0	0.0	0.6322D-02

D MATRIX

ROW 1	0.1000D+01	0.1000D-01	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 3	0.0	0.1945D+00	0.9950D+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.1000D+01	0.0	0.0	0.0	0.0	0.0				
ROW 4	0.0	0.4067D+00	0.3096D-01	0.4906D-04	-0.1519D+00	-0.1533D+00	0.2861D-01	0.0	0.0	0.0
	0.0	0.1000D+01	0.0	0.0	0.0	0.0				
ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.1000D-01	0.0	0.0	0.0	0.0
	0.0	0.0	0.1000D+01	0.0	0.0	0.0				
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.1000D+01	0.0	0.0				
ROW 7	0.0	0.6217D+01	0.4727D+00	0.0	0.0	0.0	0.3679D+00	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.1000D+01	0.0				
ROW 8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.1000D+01				
ROW 9	-0.8920D-02	0.0	0.0	-0.4080D-02	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 10	-0.5840D+00	0.0	0.0	-0.6160D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				

```
ROW 11 -0.5685D-01  0.0         0.0         -0.5996D-01  0.0         0.0         0.0         0.0         0.0         0.0
       0.0          0.0         0.0          0.0         0.0         0.0

ROW 12  0.1870D+01  0.0         0.0          0.2245D+00  0.0         0.0         0.0         0.0         0.0         0.0
       0.0          0.0         0.0          0.0         0.0         0.0

ROW 13 -0.9690D-02  0.0         0.0         -0.1725D-02  0.0         0.0         0.0         0.0         0.0         0.0
       0.0          0.0         0.0          0.0         0.0         0.0

ROW 14 -0.1938D+01  0.0         0.0         -0.3450D+00  0.0         0.0         0.0         0.0         0.0         0.0
       0.0          0.0         0.0          0.0         0.0         0.0

ROW 15 -0.2106D+01  0.0         0.0         -0.2221D+01  0.0         0.0         0.0         0.0         0.0         0.0
       0.0          0.0         0.0          0.0         0.0         0.0

ROW 16  0.0         0.0         0.0          0.0         0.0         0.0         0.0         0.0         0.0         0.0
       0.0          0.0         0.0          0.0         0.0         0.0
-------------------------------------------------------------------------------------------------------------------------
ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
        -----------------------------------
(-0.5855D+00, 0.0       ) ( 0.9044D+00, 0.4000D+00) ( 0.9044D+00,-0.4000D+00) ( 0.1906D+00, 0.0       )

( 0.9749D+00, 0.5679D-01) ( 0.9749D+00,-0.5679D-01) ( 0.9950D+00, 0.0       ) ( 0.9884D+00, 0.0       )

( 0.1588D-01, 0.0       ) (-0.2409D-15, 0.0       ) (-0.4301D-17, 0.6459D-16) (-0.4301D-17,-0.6459D-16)

(-0.1314D-17, 0.1128D-17) (-0.1314D-17,-0.1128D-17) ( 0.1000D+01, 0.0       ) ( 0.0       , 0.0       )

ALL THE EIGENVALUES OF D HAVE MODULUS LESS THAN OR EQUAL TO UNITY FOR H=0.0100 SCHEME IS STABLE FOR CHOICE OF H
*************************************************************************************************************

H=0.0110 NUMERICAL INTEGRATION STEPLENGTH IS 0.000200

        G MATRIX

ROW  1  0.1000D+01  0.1100D-01  0.0          0.0          0.0          0.0          0.0
ROW  2  0.0         0.1000D+01  0.0          0.0          0.0          0.0          0.0
ROW  3  0.0         0.2139D+00  0.9945D+00   0.0          0.0          0.0          0.0
ROW  4  0.0         0.4338D+00  0.3298D-01   0.1819D-04  -0.1519D+00  -0.1534D+00   0.2589D-01
ROW  5  0.0         0.0         0.0          0.0          0.1000D+01   0.1100D-01   0.0
ROW  6  0.0         0.0         0.0          0.0          0.0          0.1000D+01   0.0
ROW  7  0.0         0.6567D+01  0.4987D+00   0.0          0.0          0.0          0.3329D+00
-------------------------------------------------------------------------------------------------------------------------
        C MATRIX

ROW  1  0.1100D-01  0.6050D-04  0.0          0.0          0.0          0.0          0.0
ROW  2  0.0         0.1100D-01  0.0          0.0          0.0          0.0          0.0
ROW  3  0.0         0.1178D-02  0.1097D-01   0.0          0.0          0.0          0.0
ROW  4  0.0         0.2539D-02  0.1939D-03   0.1011D-02  -0.1517D-02  -0.1525D-02   0.4403D-03
```

ROW 5	0.0	0.0	0.0	0.0	0.1100D-01	0.6050D-04	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1100D-01	0.0
ROW 7	0.0	0.4245D-01	0.3240D-02	0.0	0.0	0.0	0.6672D-02

D MATRIX

ROW 1	0.1000D+01	0.1100D-01	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 3	0.0	0.2139D+00	0.9945D+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.1000D+01	0.0	0.0	0.0	0.0	0.0				
ROW 4	0.0	0.4338D+00	0.3298D-01	0.1819D-04	-0.1519D+00	-0.1534D+00	0.2589D-01	0.0	0.0	0.0
	0.0	0.1000D+01	0.0	0.0	0.0	0.0				
ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.1100D-01	0.0	0.0	0.0	0.0
	0.0	0.0	0.1000D+01	0.0	0.0	0.0				
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.1000D+01	0.0	0.0				
ROW 7	0.0	0.6567D+01	0.4987D+00	0.0	0.0	0.0	0.3329D+00	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.1000D+01	0.0				
ROW 8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.1000D+01				
ROW 9	-0.1013D-01	0.0	0.0	-0.4827D-02	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 10	-0.6424D+00	0.0	0.0	-0.6776D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 11	-0.6877D-01	0.0	0.0	-0.7254D-01	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 12	0.1876D+01	0.0	0.0	0.2039D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 13	-0.1172D-01	0.0	0.0	-0.2087D-02	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 14	-0.2132D+01	0.0	0.0	-0.3795D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 15	-0.2479D+01	0.0	0.0	-0.2615D+01	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				

ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE 0

EIGENVALUES OF D (COMPLEX ELEMENTS)

(-0.5887D+00, 0.0) (0.9020D+00, 0.4250D+00) (0.9020D+00,-0.4250D+00) (0.1660D+00, 0.0)

(0.9724D+00, 0.6262D-01) (0.9724D+00,-0.6262D-01) (0.9945D+00, 0.0) (0.9873D+00, 0.0)

(0.1960D-01, 0.0) (-0.6104D-15, 0.0) (-0.1689D-16, 0.9016D-16) (-0.1689D-16,-0.9016D-16)

(0.2419D-18, 0.1104D-17) (0.2419D-18,-0.1104D-17) (0.1000D+01, 0.0) (0.0 , 0.0)

ALL THE EIGENVALUES OF D HAVE MODULUS LESS THAN OR EQUAL TO UNITY FOR H=0.0110 SCHEME IS STABLE FOR CHOICE OF H

****

H=0.0120 NUMERICAL INTEGRATION STEPLENGTH IS 0.000200

G MATRIX

ROW 1	0.1000D+01	0.1200D-01	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.2333D+00	0.9940D+00	0.0	0.0	0.0	0.0
ROW 4	0.0	0.4585D+00	0.3482D-01	0.6743D-05	-0.1519D+00	-0.1536D+00	0.2343D-01
ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.1200D-01	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
ROW 7	0.0	0.6886D+01	0.5222D+00	0.0	0.0	0.0	0.3012D+00

Q MATRIX

ROW 1	0.1200D-01	0.7200D-04	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.1200D-01	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.1401D-02	0.1196D-01	0.0	0.0	0.0	0.0
ROW 4	0.0	0.2985D-02	0.2278D-03	0.1011D-02	-0.1669D-02	-0.1678D-02	0.4649D-03
ROW 5	0.0	0.0	0.0	0.0	0.1200D-01	0.7200D-04	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1200D-01	0.0
ROW 7	0.0	0.4918D-01	0.3750D-02	0.0	0.0	0.0	0.6989D-02

D MATRIX

ROW 1	0.1000D+01	0.1200D-01	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 3	0.0	0.2333D+00	0.9940D+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.1000D+01	0.0	0.0	0.0	0.0	0.0				
ROW 4	0.0	0.4585D+00	0.3482D-01	0.6743D-05	-0.1519D+00	-0.1536D+00	0.2343D-01	0.0	0.0	0.0
	0.0	0.1000D+01	0.0	0.0	0.0	0.0				
ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.1200D-01	0.0	0.0	0.0	0.0
	0.0	0.0	0.1000D+01	0.0	0.0	0.0				

ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.1000D+01	0.0	0.0				
ROW 7	0.0	0.6886D+01	0.5222D+00	0.0	0.0	0.0	0.3012D+00	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.1000D+01	0.0				
ROW 8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.1000D+01				
ROW 9	-0.1140D-01	0.0	0.0	-0.5635D-02	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 10	-0.7068D+00	0.0	0.0	-0.7392D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 11	-0.8163D-01	0.0	0.0	-0.8631D-01	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 12	0.1879D+01	0.0	0.0	0.1817D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 13	-0.1395D-01	0.0	0.0	-0.2484D-02	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 14	-0.2326D+01	0.0	0.0	-0.4140D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 15	-0.2872D+01	0.0	0.0	-0.3030D+01	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				

ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE 0

EIGENVALUES OF D (COMPLEX ELEMENTS)

(-0.5916D+00, 0.0)	(0.8997D+00, 0.4489D+00)	(0.8997D+00,-0.4489D+00)	(0.1436D+00, 0.0)
(0.9698D+00, 0.6849D-01)	(0.9698D+00,-0.6849D-01)	(0.9940D+00, 0.0)	(0.9861D+00, 0.0)
(0.2398D-01, 0.0)	(-0.6030D-15, 0.0)	(0.1582D-16, 0.9646D-16)	(0.1582D-16,-0.9646D-16)
(-0.1552D-16, 0.0)	(-0.1020D-17, 0.0)	(0.1000D+01, 0.0)	(0.0 , 0.0)

FOR H=0.0120, 2 OF THE EIGENVALUES OF D HAVE MODULUS GREATER THAN UNITY.THEREFORE CHOICE IS UNSTABLE FOR THIS H
(0.8997D+00, 0.4489D+00) (0.8997D+00,-0.4489D+00) (

****

H=0.0130 NUMERICAL INTEGRATION STEPLENGTH IS 0.000200

G MATRIX

ROW 1	0.1000D+01	0.1300D-01	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.2527D+00	0.9935D+00	0.0	0.0	0.0	0.0
ROW 4	0.0	0.4809D+00	0.3647D-01	0.2500D-05	-0.1519D+00	-0.1537D+00	0.2120D-01

ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.1300D-01	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
ROW 7	0.0	0.7176D+01	0.5434D+00	0.0	0.0	0.0	0.2726D+00

C MATRIX

ROW 1	0.1300D-01	0.8450D-04	0.0	0.0	0.0	0.0	0.0
ROW 2	0.0	0.1300D-01	0.0	0.0	0.0	0.0	0.0
ROW 3	0.0	0.1644D-02	0.1296D-01	0.0	0.0	0.0	0.0
ROW 4	0.0	0.3455D-02	0.2634D-03	0.1011D-02	-0.1821D-02	-0.1832D-02	0.4872D-03
ROW 5	0.0	0.0	0.0	0.0	0.1300D-01	0.8450D-04	0.0
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1300D-01	0.0
ROW 7	0.0	0.5621D-01	0.4283D-02	0.0	0.0	0.0	0.7275D-02

D MATRIX

ROW 1	0.1000D+01	0.1300D-01	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 2	0.0	0.1000D+01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 3	0.0	0.2527D+00	0.9935D+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.1000D+01	0.0	0.0	0.0	0.0	0.0				
ROW 4	0.0	0.4809D+00	0.3647D-01	0.2500D-05	-0.1519D+00	-0.1537D+00	0.2120D-01	0.0	0.0	0.0
	0.0	0.1000D+01	0.0	0.0	0.0	0.0				
ROW 5	0.0	0.0	0.0	0.0	0.1000D+01	0.1300D-01	0.0	0.0	0.0	0.0
	0.0	0.0	0.1000D+01	0.0	0.0	0.0				
ROW 6	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.1000D+01	0.0	0.0				
ROW 7	0.0	0.7176D+01	0.5434D+00	0.0	0.0	0.0	0.2726D+00	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.1000D+01	0.0				
ROW 8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1000D+01	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.1000D+01				
ROW 9	-0.1273D-01	0.0	0.0	-0.6505D-02	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 10	-0.7592D+00	0.0	0.0	-0.8008D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 11	-0.9602D-01	0.0	0.0	-0.1013D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 12	0.1882D+01	0.0	0.0	0.1581D+00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0				
ROW 13	-0.1638D-01	0.0	0.0	-0.2915D-02	0.0	0.0	0.0	0.0	0.0	0.0

```
          0.0          0.0          0.0          0.0          0.0          0.0

ROW 14 -0.2519D+01   0.0          0.0         -0.4485D+00   0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0

ROW 15 -0.3283D+01   0.0          0.0         -0.3463D+01   0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0

ROW 16   0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0
-------------------------------------------------------------------------------------------------------------------------------------
ERROR PARAMETER FROM EIGENVALUE ROUTINE HAS THE VALUE   0

        EIGENVALUES OF D (COMPLEX ELEMENTS)
        -----------------------------------

(-0.5939D+00, 0.0       )  ( 0.8974D+00, 0.4716D+00)  ( 0.8974D+00,-0.4716D+00)  ( 0.1227D+00, 0.0       )

( 0.9673D+00, 0.7438D-01)  ( 0.9673D+00,-0.7438D-01)  ( 0.9935D+00, 0.0       )  ( 0.9850D+00, 0.0       )

( 0.2932D-01, 0.0       )  (-0.1284D-15, 0.5338D-16)  (-0.1284D-15,-0.5338D-16)  ( 0.4292D-16, 0.0       )

(-0.1135D-16, 0.0       )  ( 0.1051D-17, 0.0       )  ( 0.1000D+01, 0.0       )  ( 0.0       , 0.0       )

FOR H=0.0130,  2 OF THE EIGENVALUES OF D HAVE MODULUS GREATER THAN UNITY.THEREFORE CHOICE IS UNSTABLE FOR THIS H
( 0.8974D+00, 0.4716D+00)  ( 0.8974D+00,-0.4716D+00)  (
*************************************************************************************************************************************
```

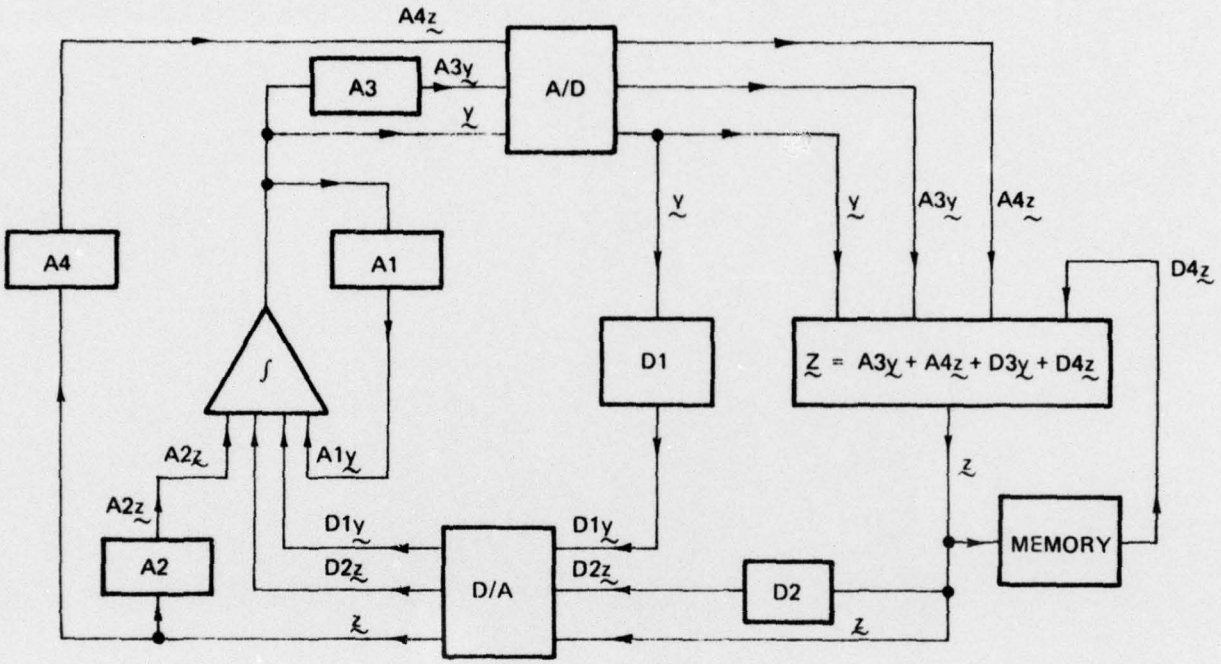


Figure 1. Partitioning scheme for general solution

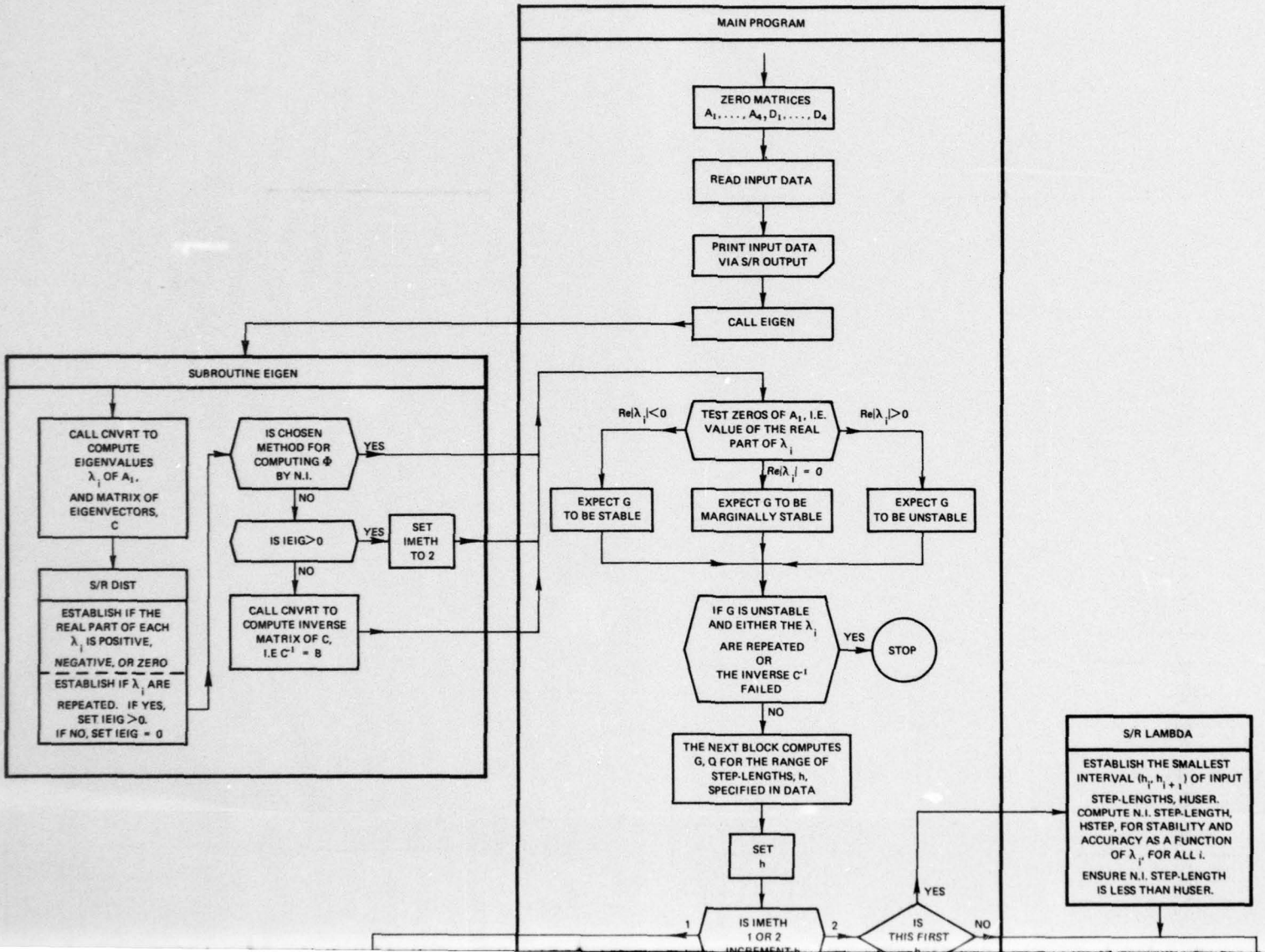
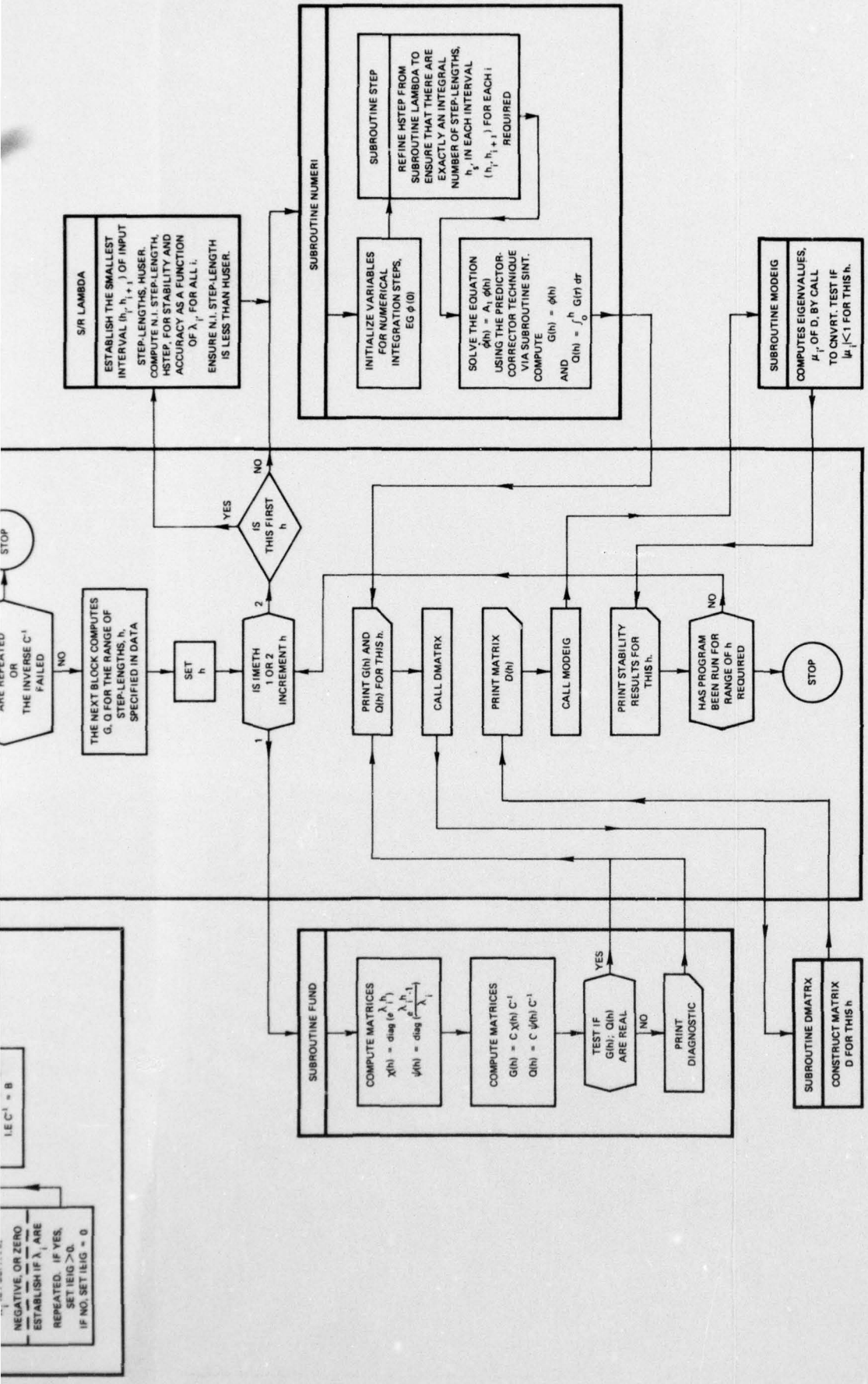


Figure 2. Flowchart of the sc



e 2. Flowchart of the software package

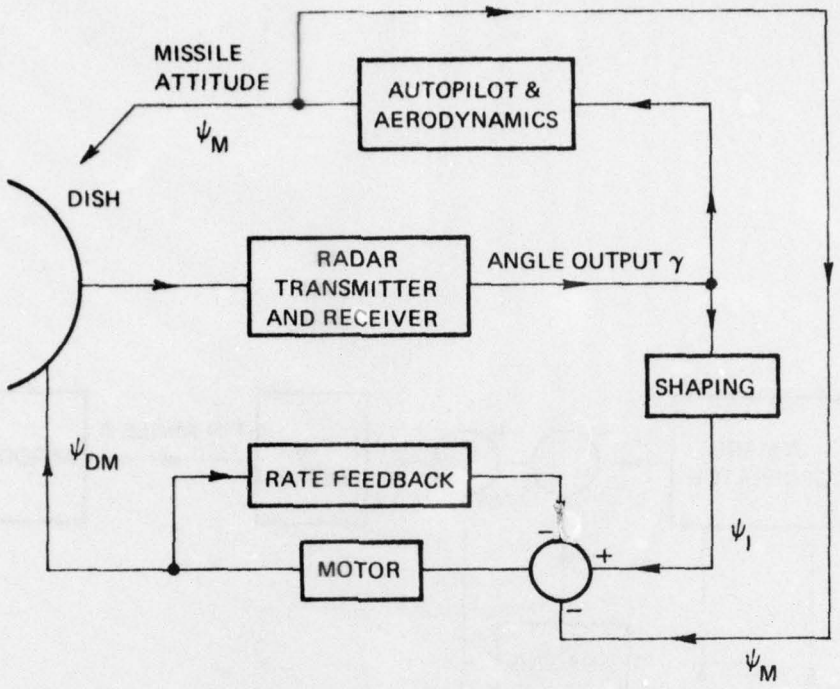


Figure 3. Block diagram of tracking radar simulation

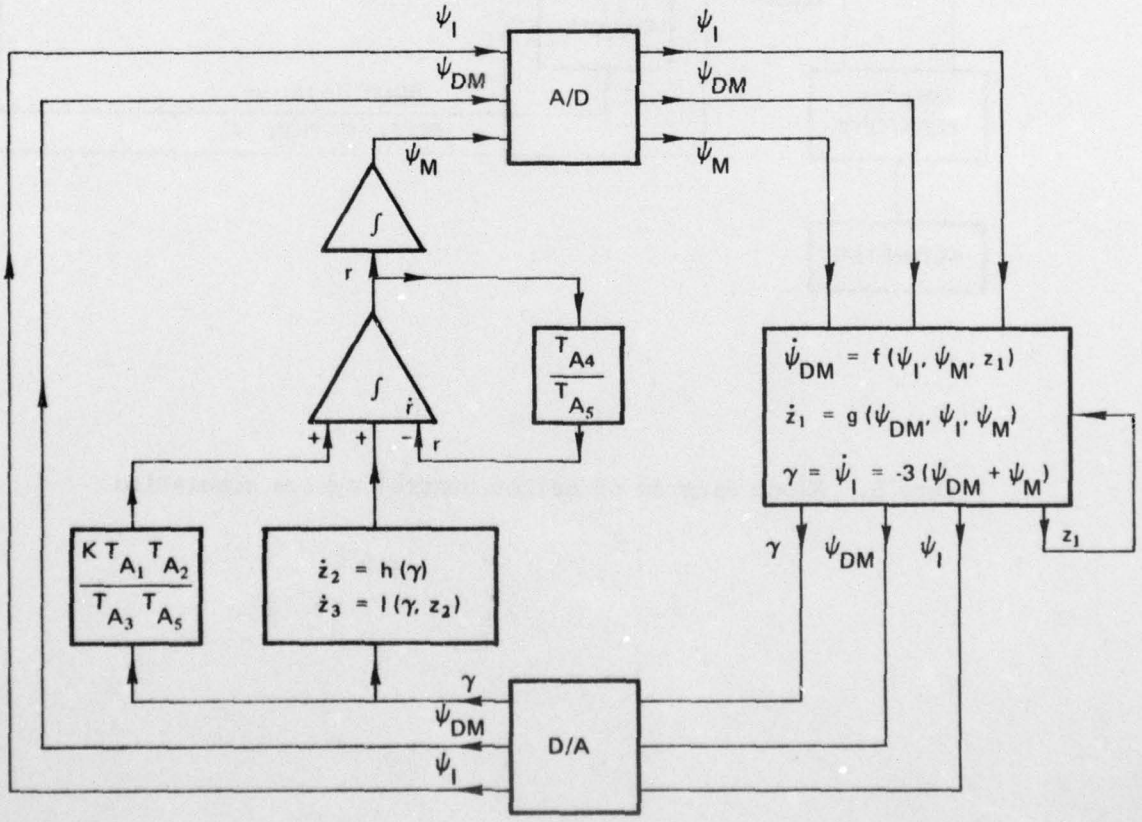


Figure 4. Partitioning scheme for tracking radar simulation

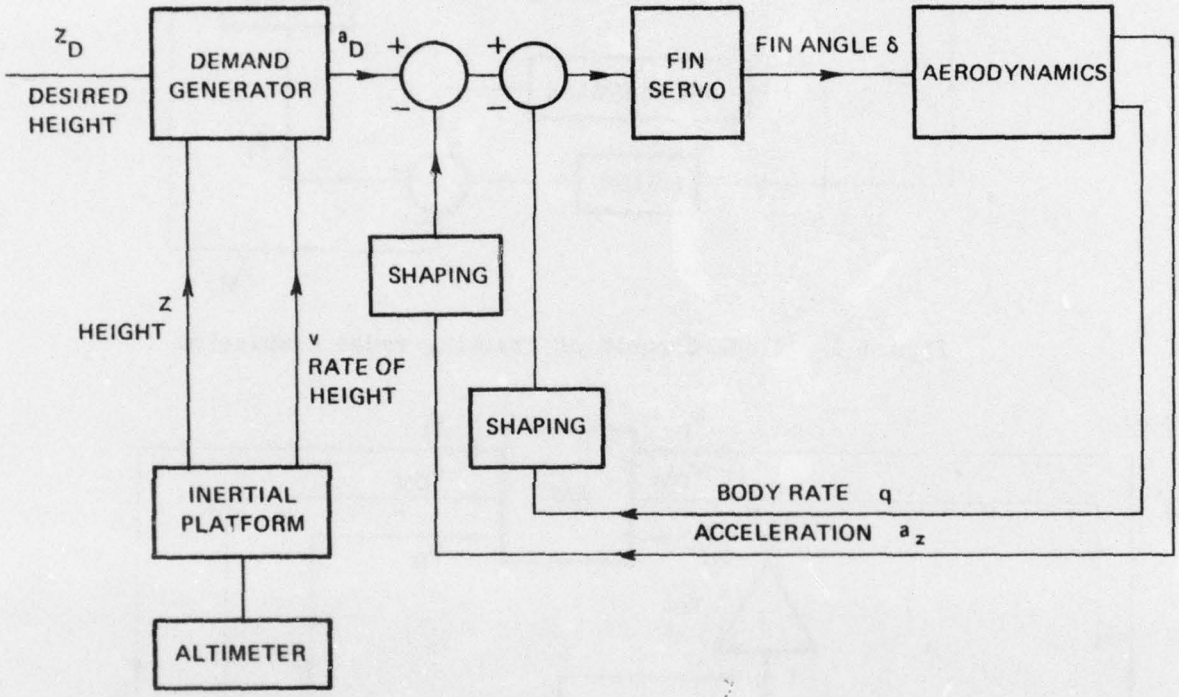


Figure 5. Block diagram of height control system simulation

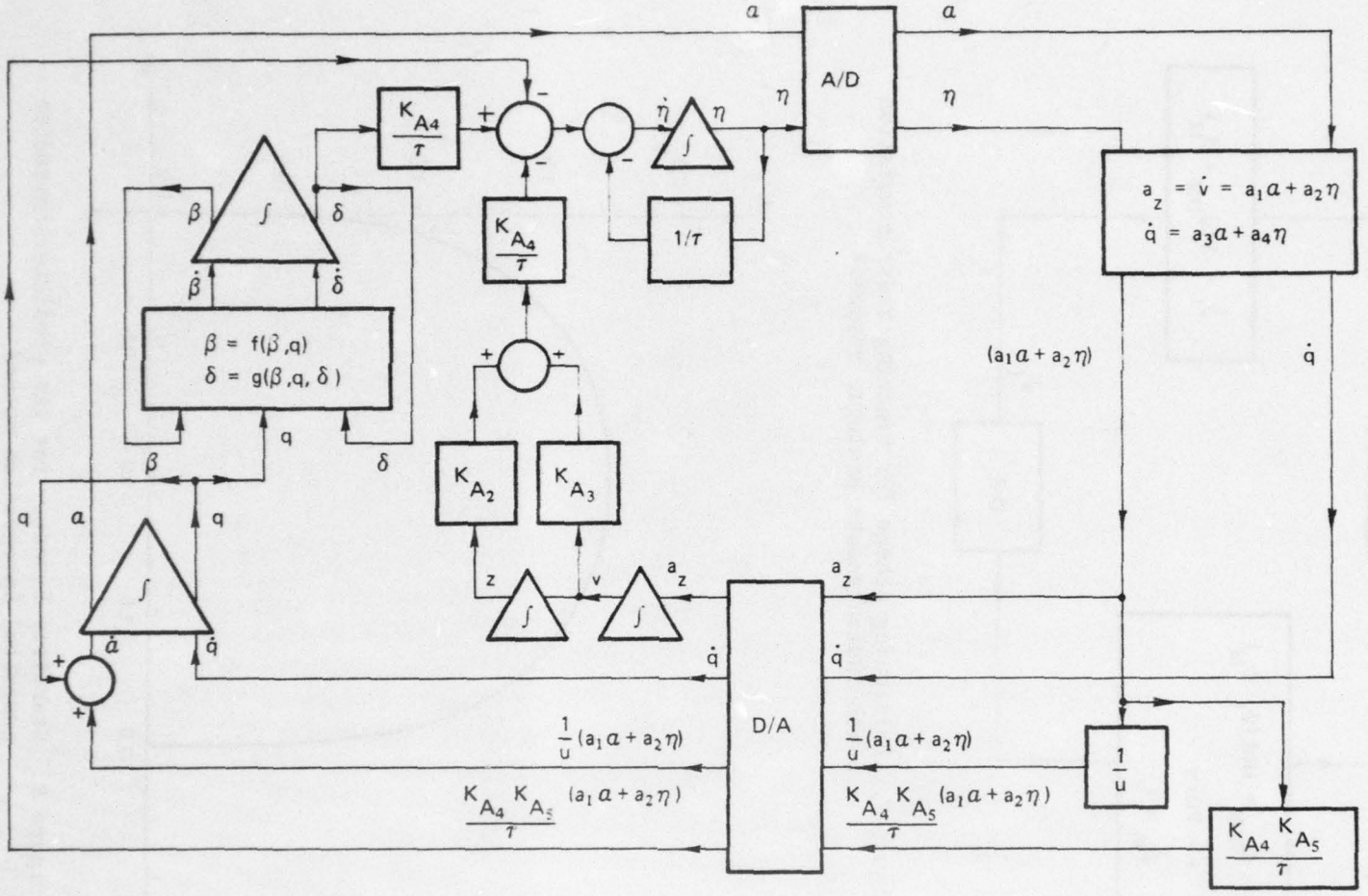


Figure 6. Partitioning scheme for height control system simulation

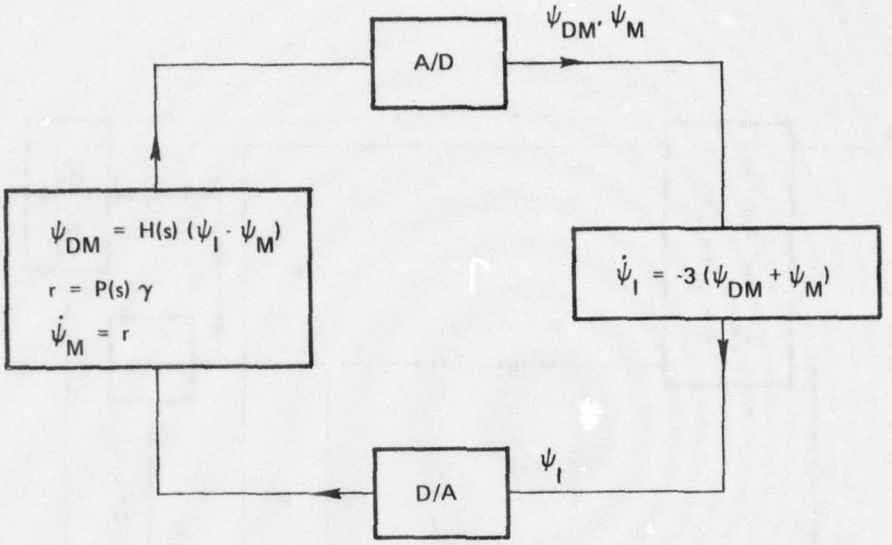


Figure 7. Partioning scheme for tracking radar simulation with instantaneous analogue response

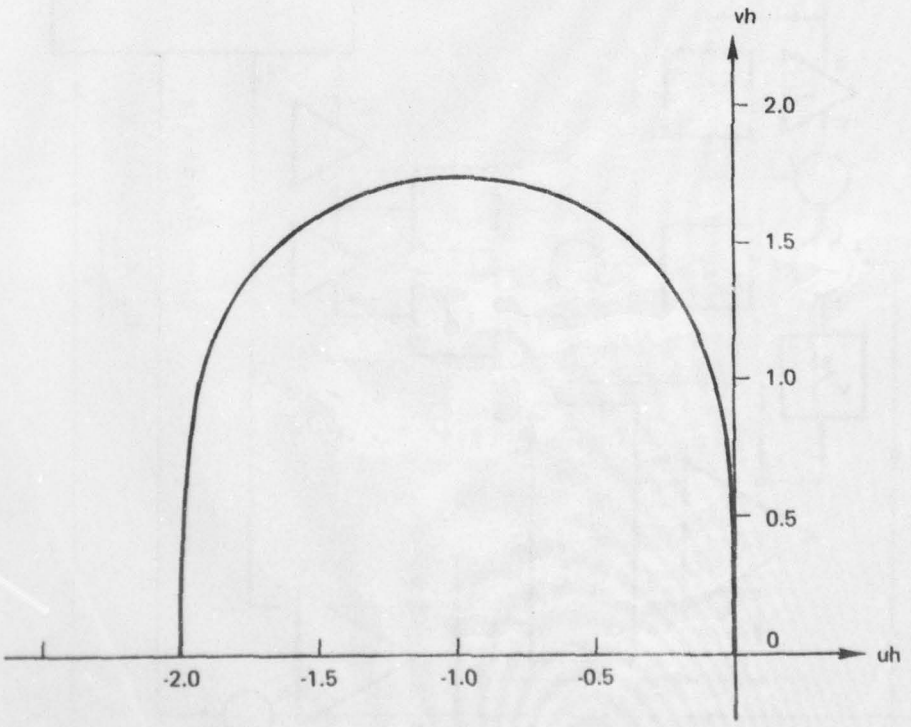


Figure 8. Stability boundary for the predictor-corrector numerical integration method

END
1-78
DDC